**Run Length Encoding之**

**電路設計與後端**

**Digital-Based Design Flow**

**流程實作**

**學號:M16131111**

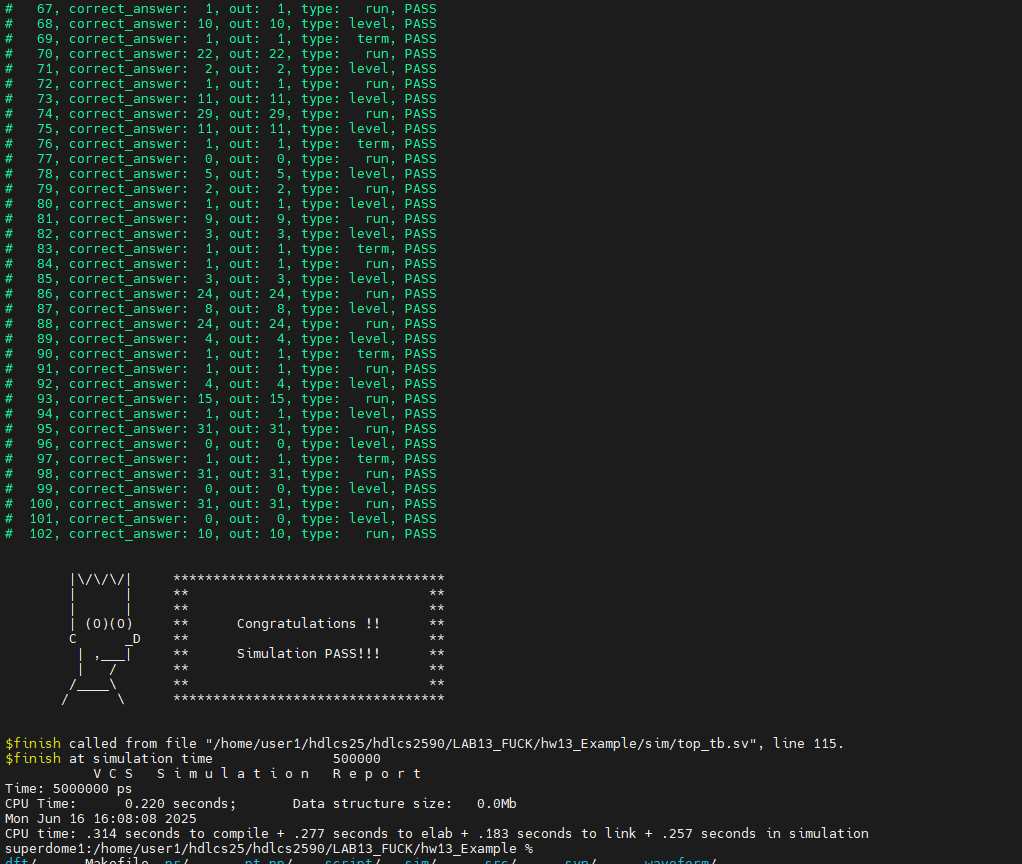
**姓名:童品綸**

1. 電路功能介紹與應用:

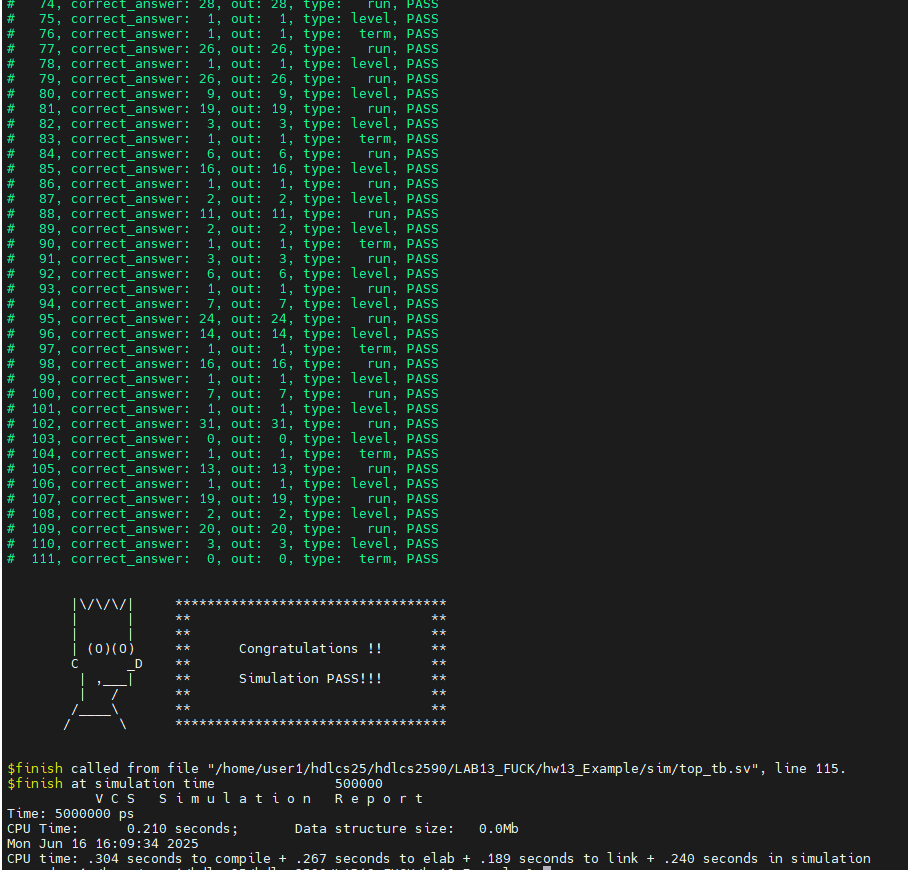
此次電路是RLE，是一種可變長度編碼，可以讓資料有大量0時，大幅降低資料symbol數量，再會出現大量0的場景，如人工智慧模型中經過ReLU運算或權重剪枝後，資料會出現大量的0，因此可以在傳回DRAM時可以經過編碼，從DRAM讀出來後再進行解碼，如此一來可以大幅降低與DRAM傳輸之資料量。

電路功能驗證:

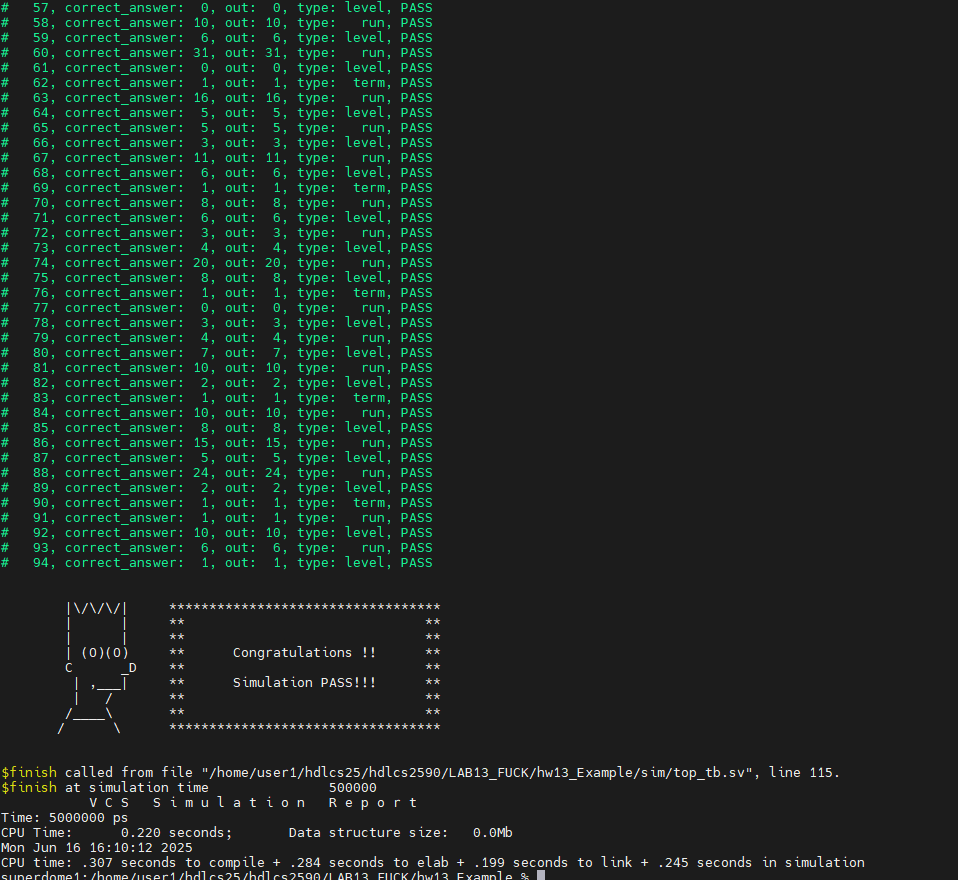
RTL\_SIM TEST=0:



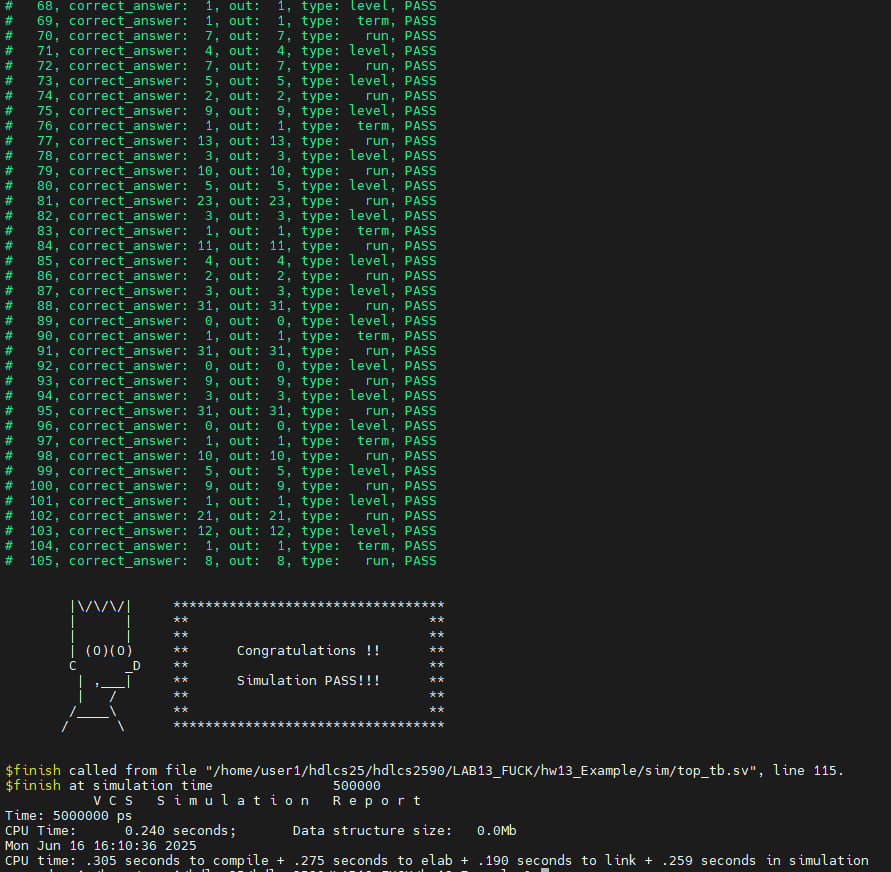
RTL\_SIM TEST=1:



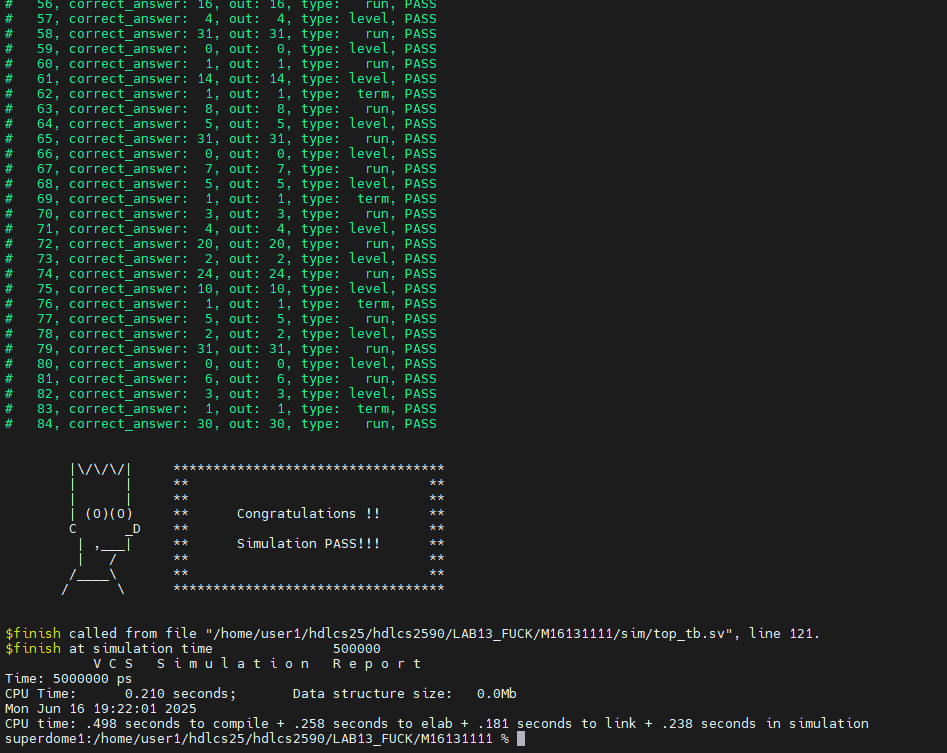
RTL\_SIM TEST=2:



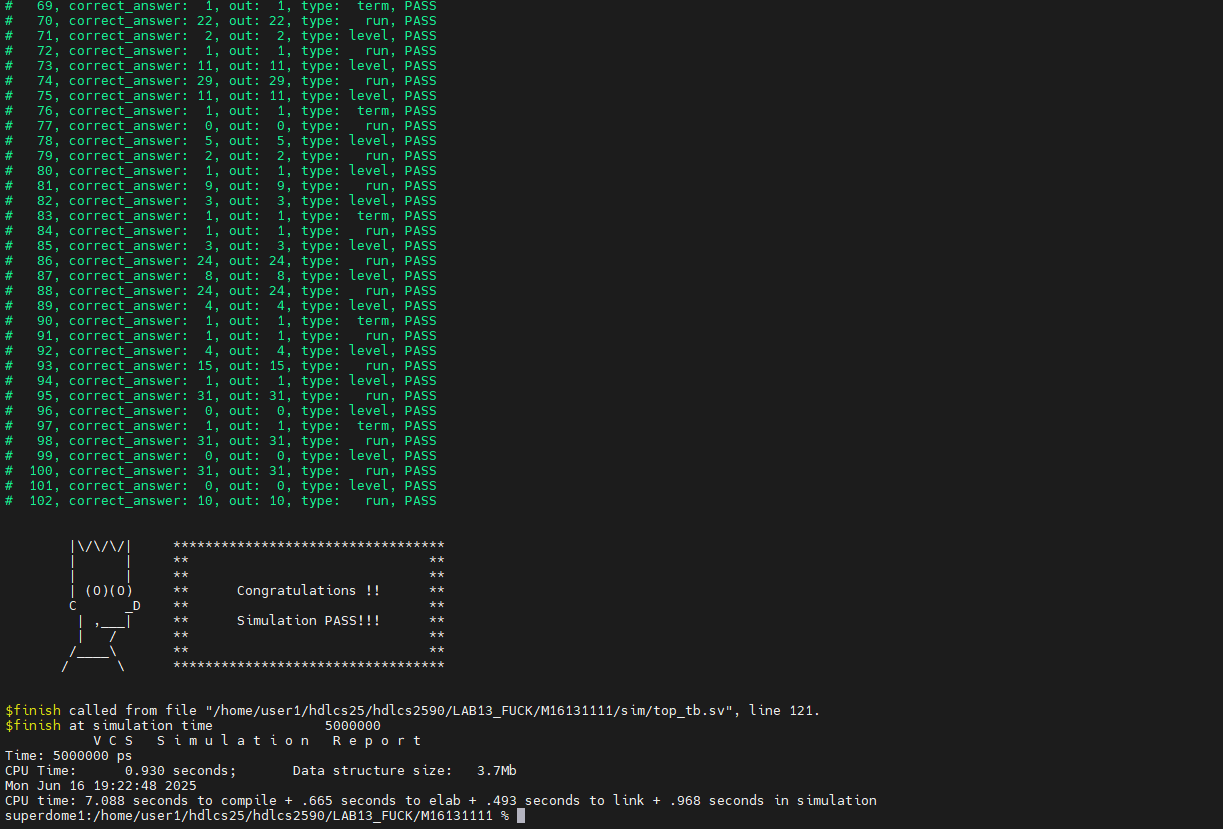
RTL\_SIM TEST=3:



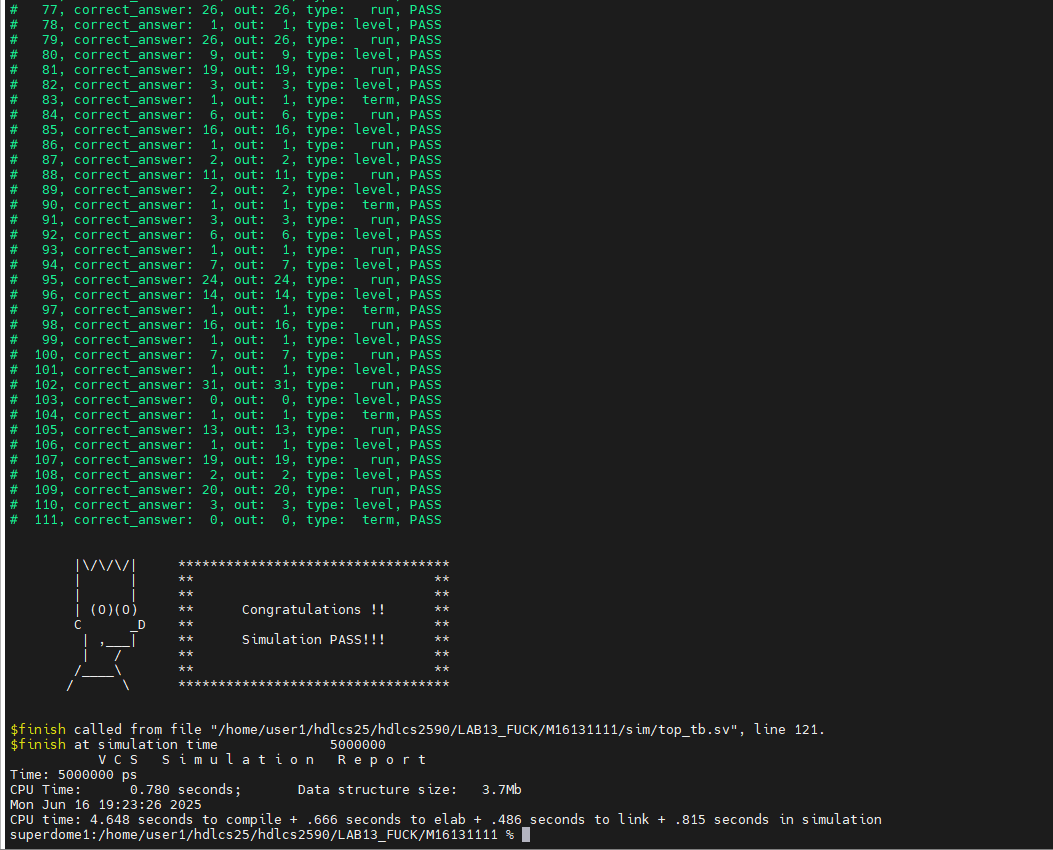
RTL\_SIM TEST=4



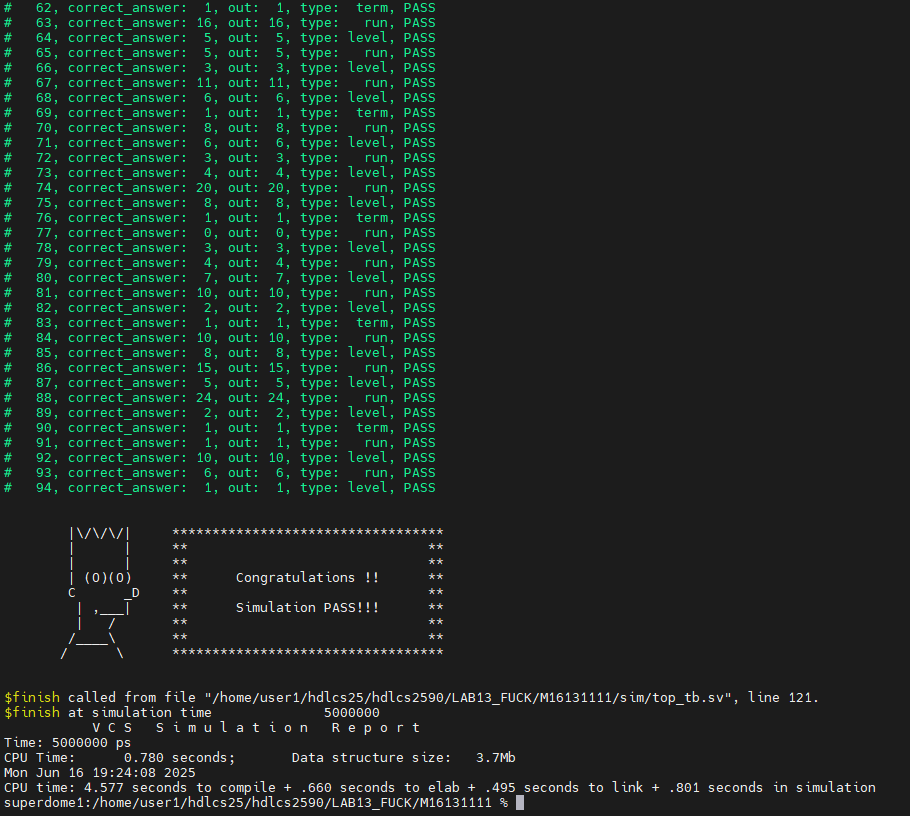
POST\_SYN\_SIM TEST=0



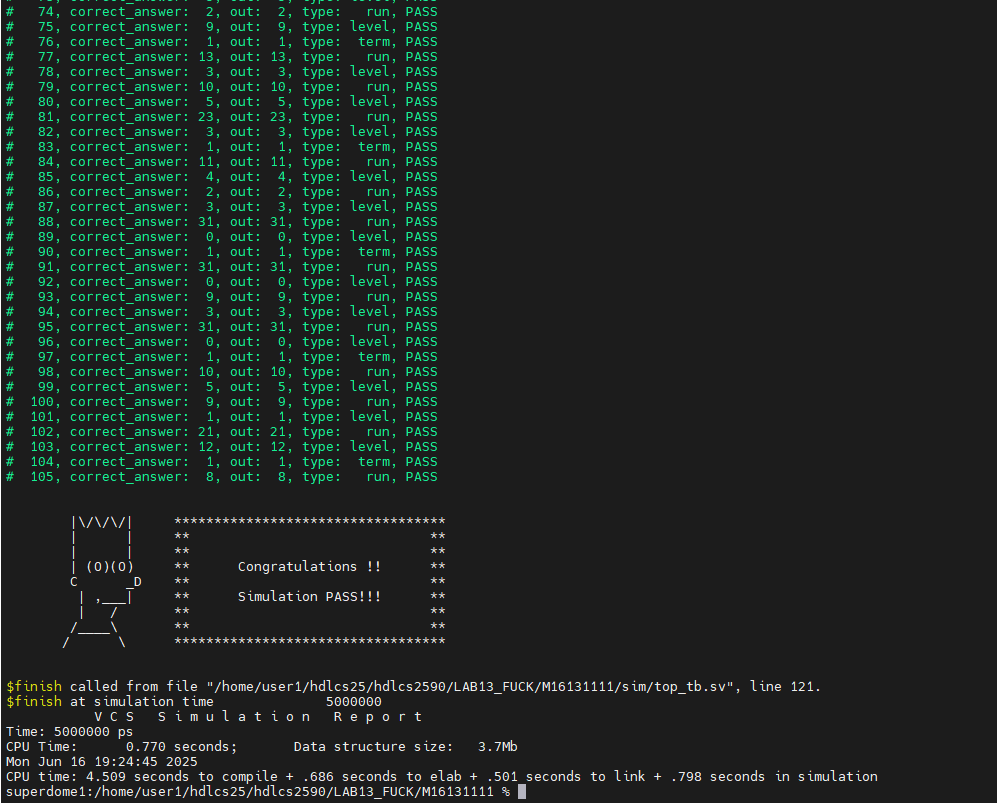
POST\_SYN\_SIM TEST=1



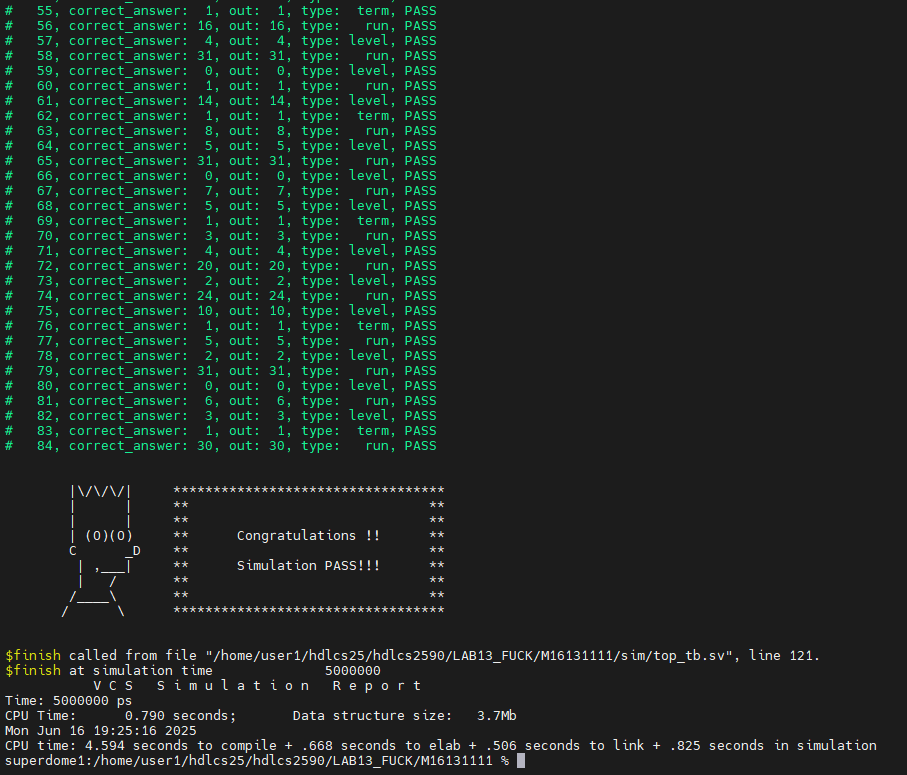
POST\_SYN\_SIM TEST=2



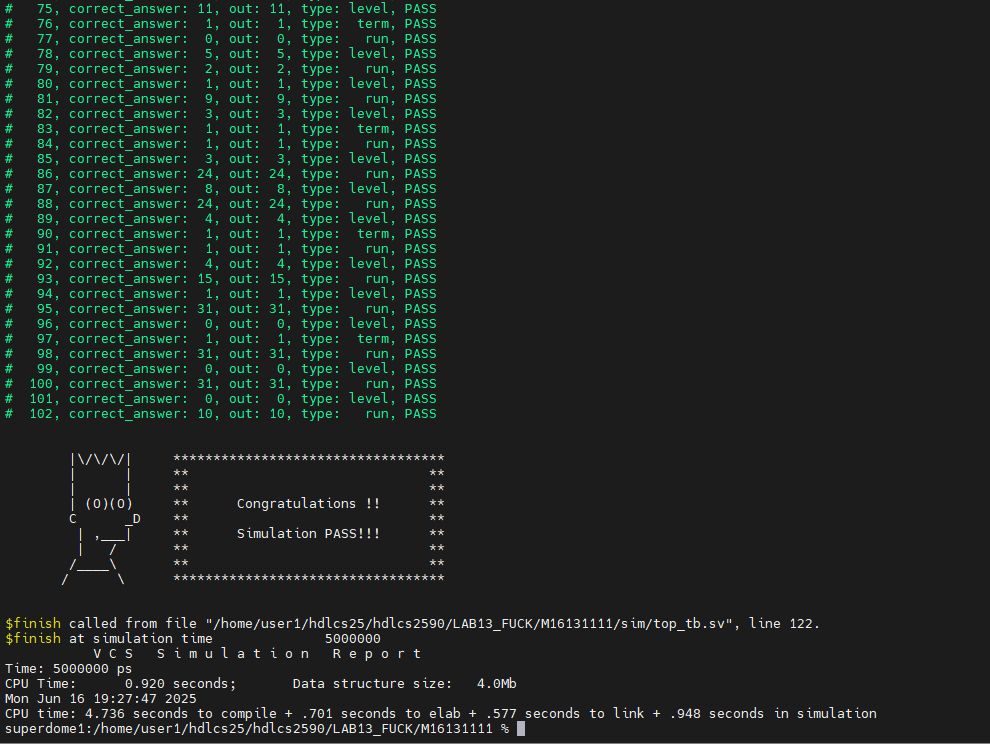
POST\_SYN\_SIM TEST=3



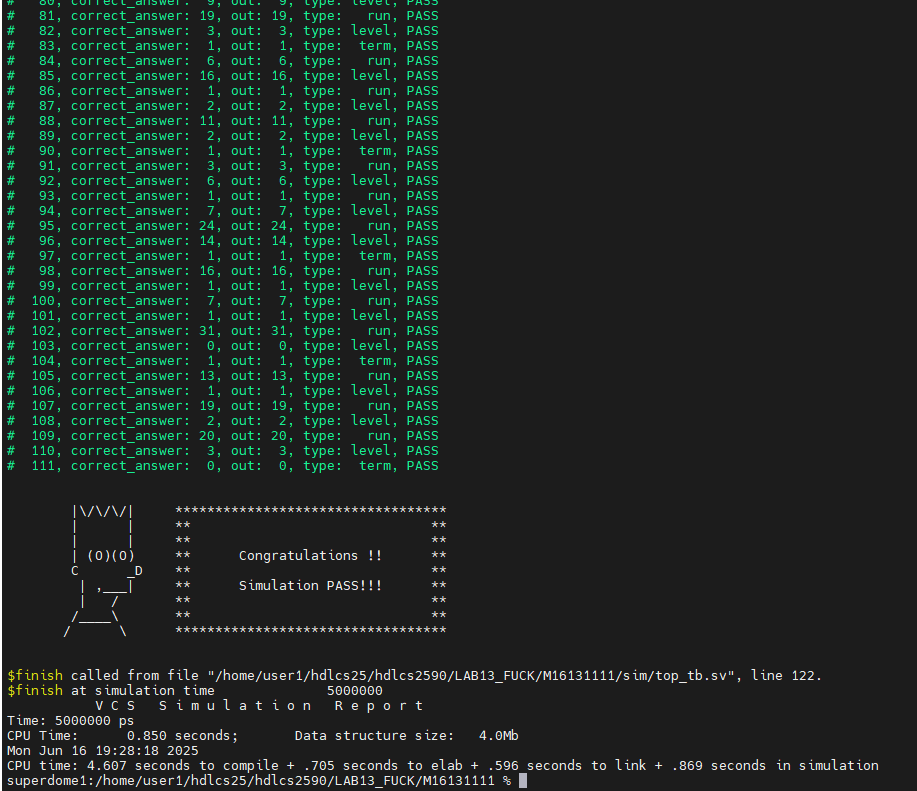
POST\_SYN\_SIM TEST=4



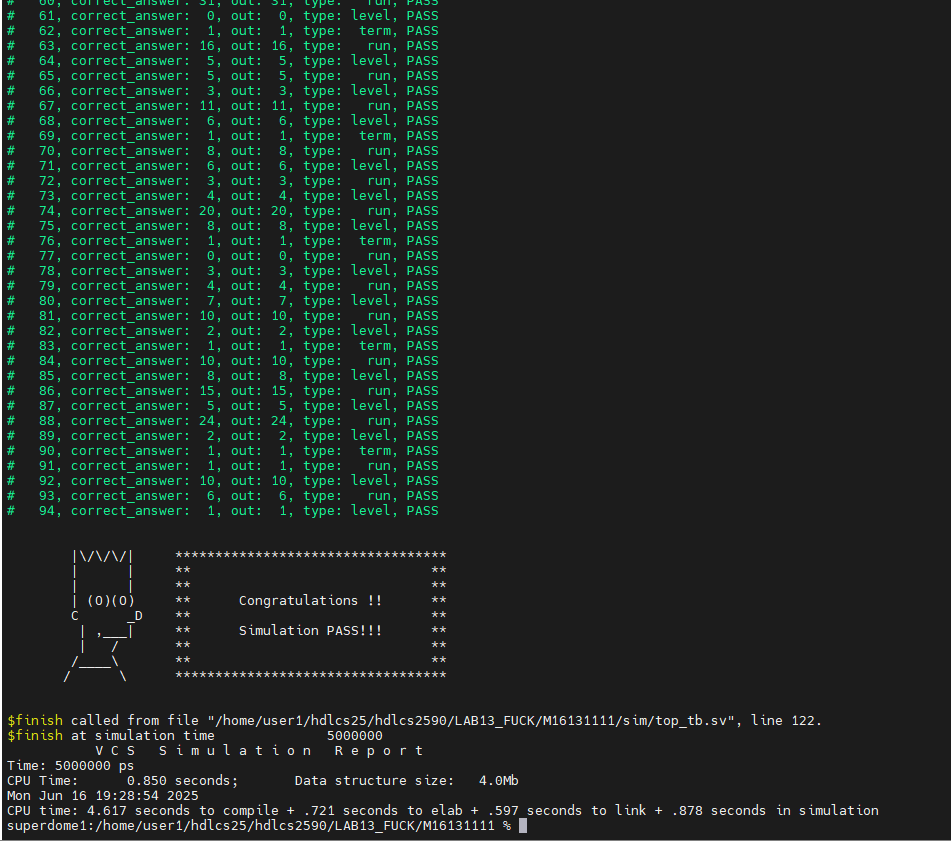
POST\_LAYOUT\_SUM TEST=0



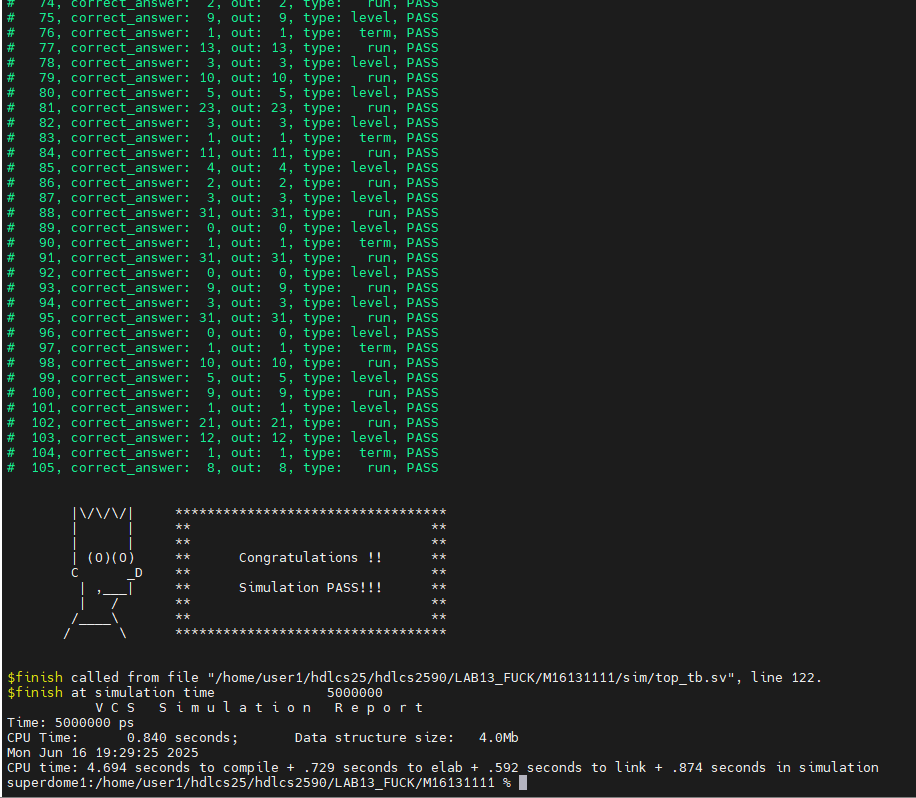
POST\_LAYOUT\_SUM TEST=1



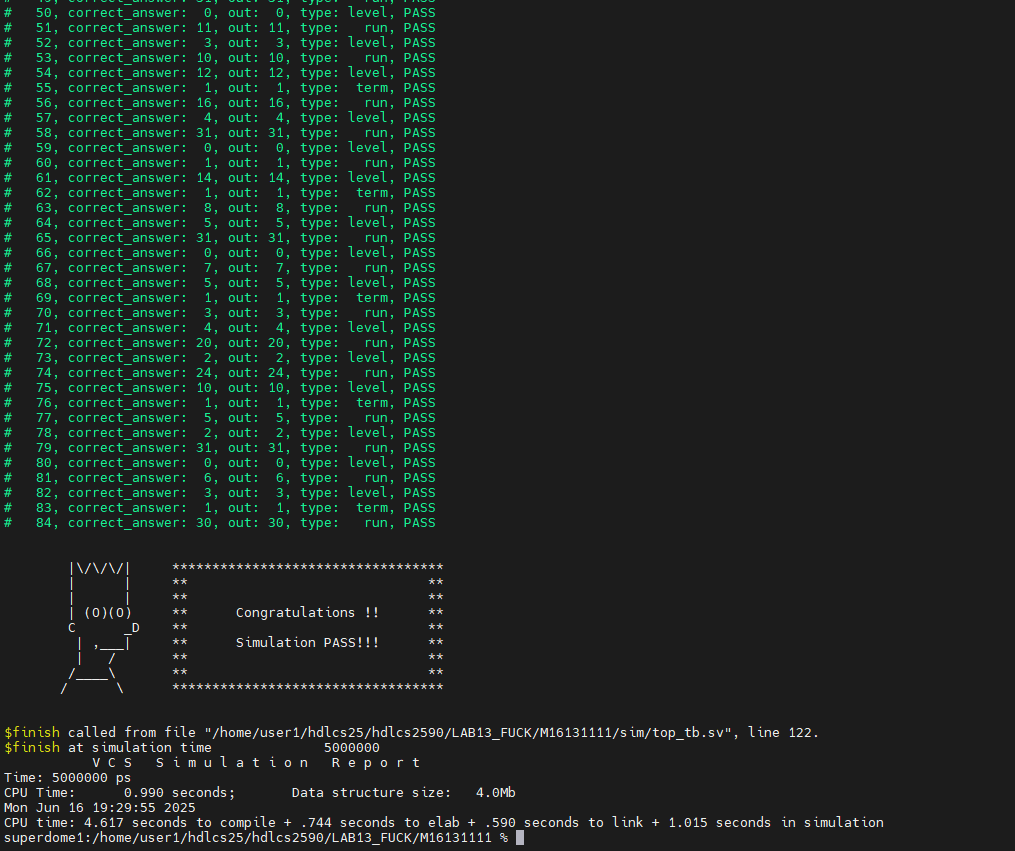
POST\_LAYOUT\_SUM TEST=2



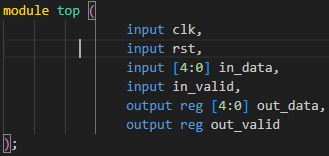
POST\_LAYOUT\_SUM TEST=3

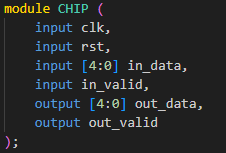


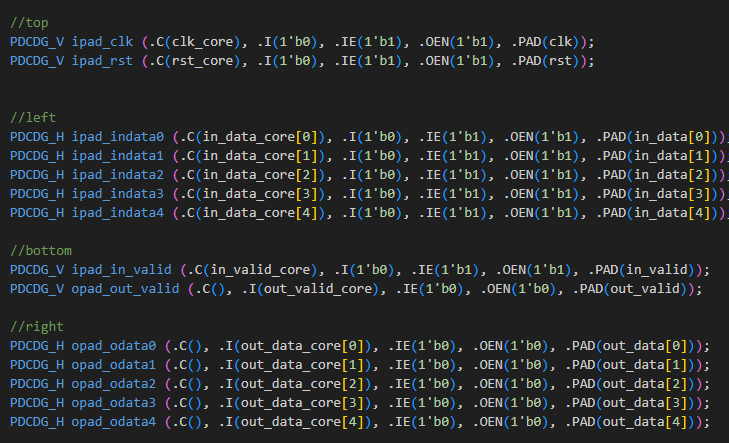
POST\_LAYOUT\_SUM TEST=4



1. IN/OUT Pin/Port宣告 :







1. Design Compiler

Design constraint

set clk\_period 4.0

create\_clock -name clk -period $clk\_period [get\_ports clk]

set\_dont\_touch\_network [all\_clocks]

set\_fix\_hold [all\_clocks]

set\_clock\_uncertainty 0.1 [all\_clocks]

set\_clock\_latency 0.5 [all\_clocks]

set\_input\_transition 0.2 [all\_inputs]

set\_clock\_transition 0.1 [all\_clocks]

set\_ideal\_network [all\_clocks]

set\_dont\_touch [get\_cells ipad\_\*]

set\_dont\_touch [get\_cells opad\_\*]

set\_operating\_conditions -min\_library N16ADFP\_StdCellff0p88v125c -min ff0p88v125c \

-max\_library N16ADFP\_StdCellss0p72vm40c -max ss0p72vm40c

set\_dont\_use N16ADFP\_StdCellss0p72vm40c/TIEHBWP16P90

set\_dont\_use N16ADFP\_StdCellss0p72vm40c/TIEHBWP16P90LVT

set\_dont\_use N16ADFP\_StdCellss0p72vm40c/TIEHBWP20P90

set\_dont\_use N16ADFP\_StdCellss0p72vm40c/TIEHBWP20P90LVT

set\_dont\_use N16ADFP\_StdCellss0p72vm40c/TIELBWP16P90

set\_dont\_use N16ADFP\_StdCellss0p72vm40c/TIELBWP16P90LVT

set\_dont\_use N16ADFP\_StdCellss0p72vm40c/TIELBWP20P90

set\_dont\_use N16ADFP\_StdCellss0p72vm40c/TIELBWP20P90LVT

N16ADFP\_StdCellff0p88vm40c/TIEHBWP16P90LVT

N16ADFP\_StdCellff0p88vm40c/TIEHBWP20P90LVT

N16ADFP\_StdCellff0p88vm40c/TIELBWP16P90LVT

N16ADFP\_StdCellff0p88vm40c/TIELBWP20P90LVT

set\_dont\_use N16ADFP\_StdCellff0p88v125c/TIEHBWP16P90

set\_dont\_use N16ADFP\_StdCellff0p88v125c/TIEHBWP16P90LVT

set\_dont\_use N16ADFP\_StdCellff0p88v125c/TIEHBWP20P90

set\_dont\_use N16ADFP\_StdCellff0p88v125c/TIEHBWP20P90LVT

set\_dont\_use N16ADFP\_StdCellff0p88v125c/TIELBWP16P90

set\_dont\_use N16ADFP\_StdCellff0p88v125c/TIELBWP16P90LVT

set\_dont\_use N16ADFP\_StdCellff0p88v125c/TIELBWP20P90

set\_dont\_use N16ADFP\_StdCellff0p88v125c/TIELBWP20P90LVT

set input\_max\_clk [expr {double(round(1000\*$clk\_period \* 0.6))/1000}]

set input\_min\_clk [expr {double(round(1000\*$clk\_period \* 0.0))/1000}]

set output\_max\_clk [expr {double(round(1000\*$clk\_period \* 0.1))/1000}]

set output\_min\_clk [expr {double(round(1000\*$clk\_period \* 0.0))/1000}]

set\_load [load\_of "N16ADFP\_StdIOss0p72v1p62v125c/PDCDG\_H/PAD"] [all\_outputs]

set\_load [load\_of "N16ADFP\_StdIOss0p72v1p62v125c/PDCDG\_V/PAD"] [all\_outputs]

set\_driving\_cell -library N16ADFP\_StdIOss0p72v1p62v125c -lib\_cell PDCDG\_H -pin {C} [all\_inputs]

set\_driving\_cell -library N16ADFP\_StdIOss0p72v1p62v125c -lib\_cell PDCDG\_V -pin {C} [all\_inputs]

set auto\_wire\_load\_selection

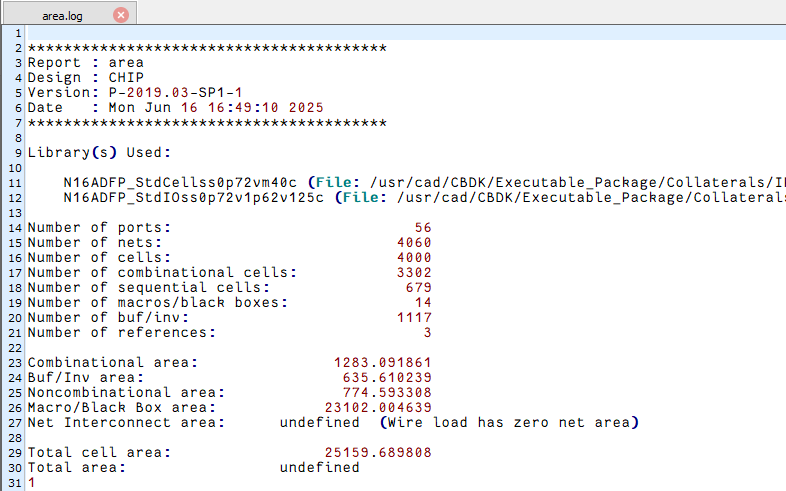
set\_wire\_load\_model -name ZeroWireload -library N16ADFP\_StdCellss0p72vm40c

set\_max\_fanout 10 [all\_inputs]

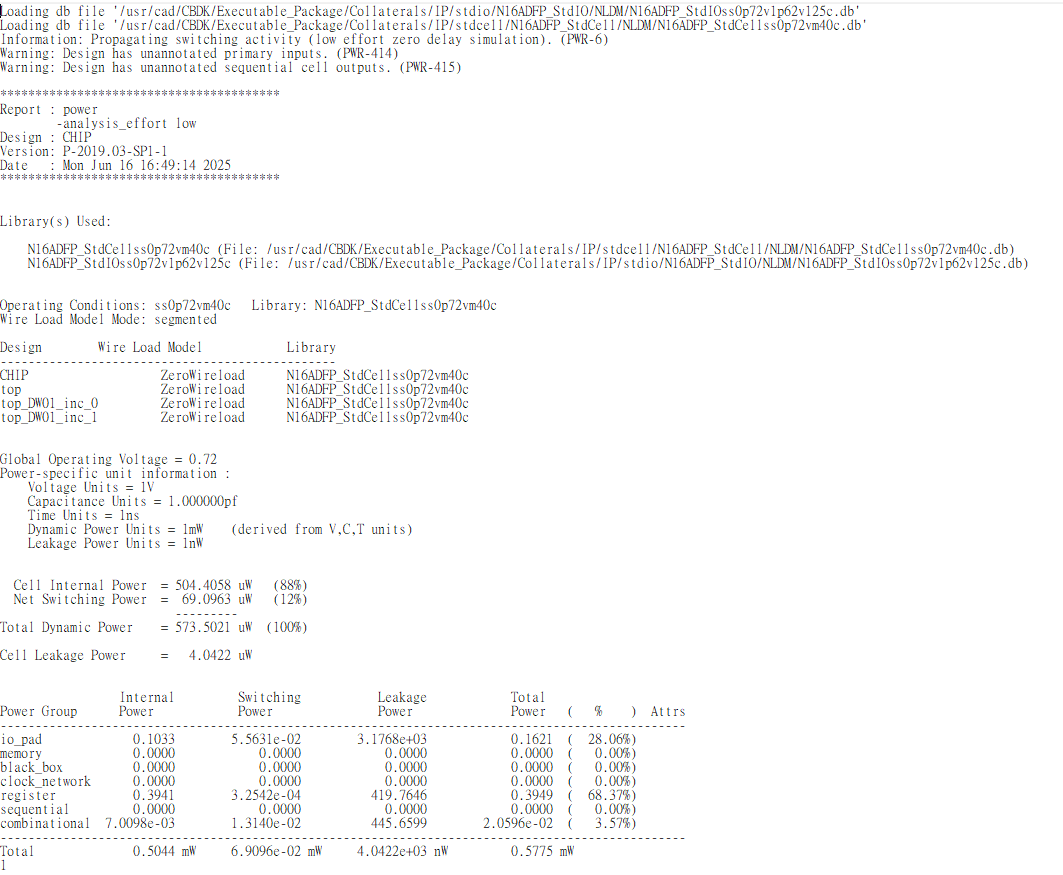
set\_max\_transition 0.1 [all\_inputs]

set\_max\_capacitance 0.1 [all\_inputs]

Area report

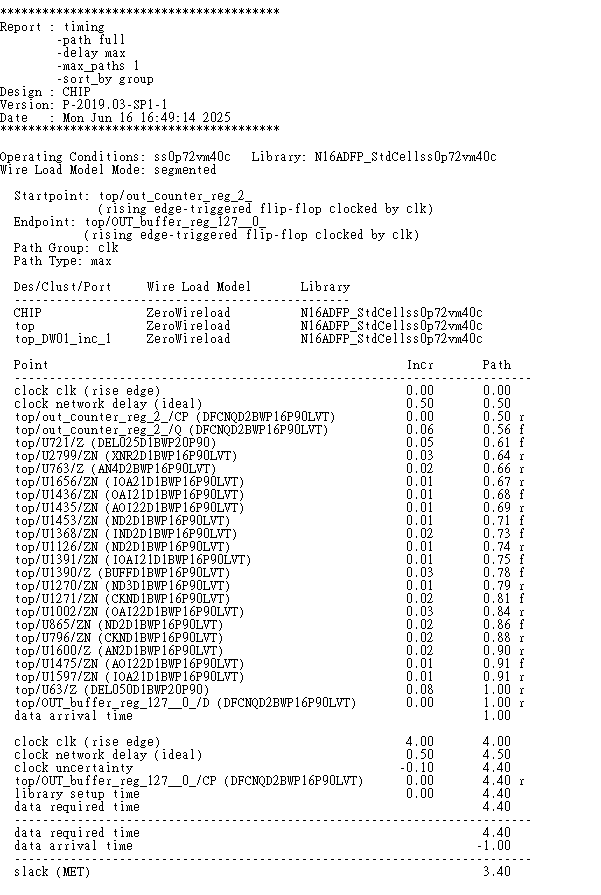


Power report

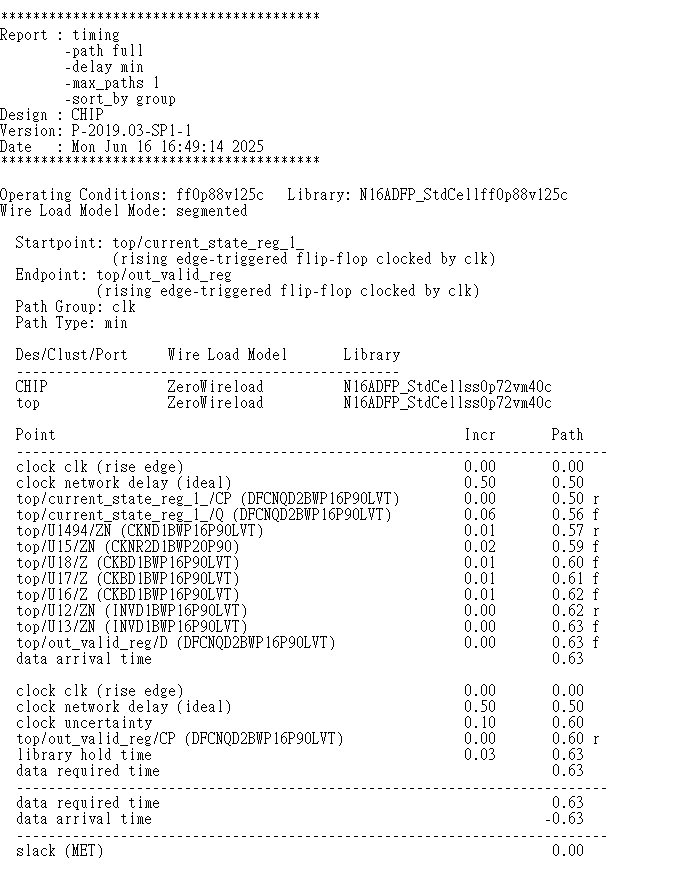


Timing report

Set up:



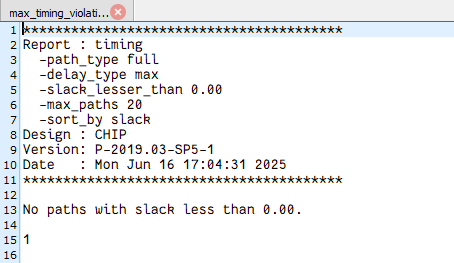
Hold time:



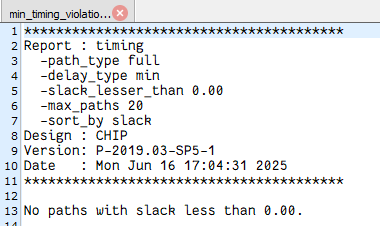
1. Post-Synthesis Timing/Power Report :

Primetime report

Set up time violation path:

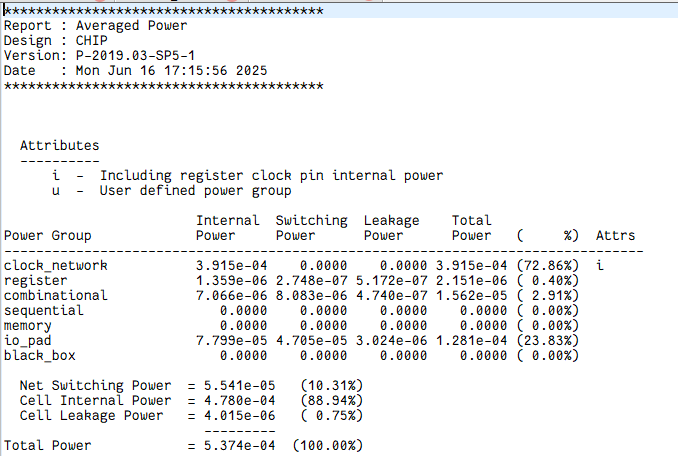


Hold time violation path:

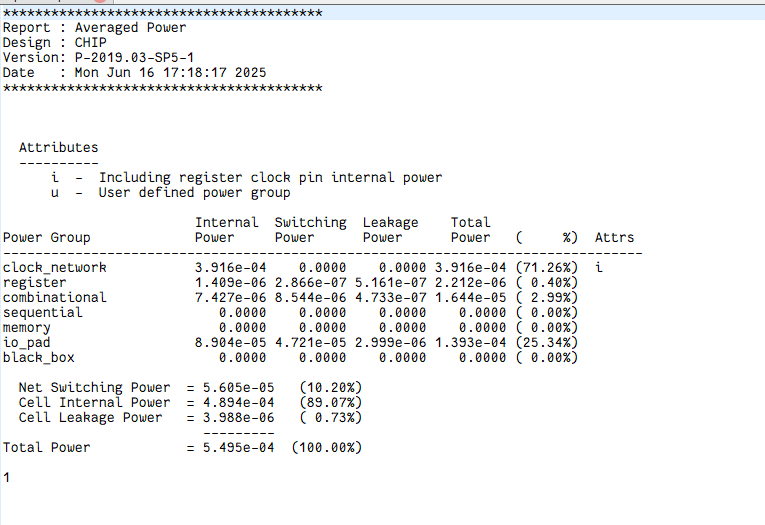


Primepower report

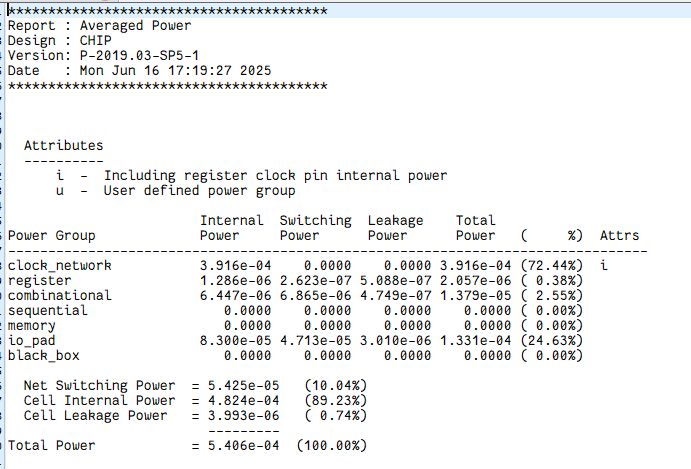
根據pattern 0的toggle情形之power計算:



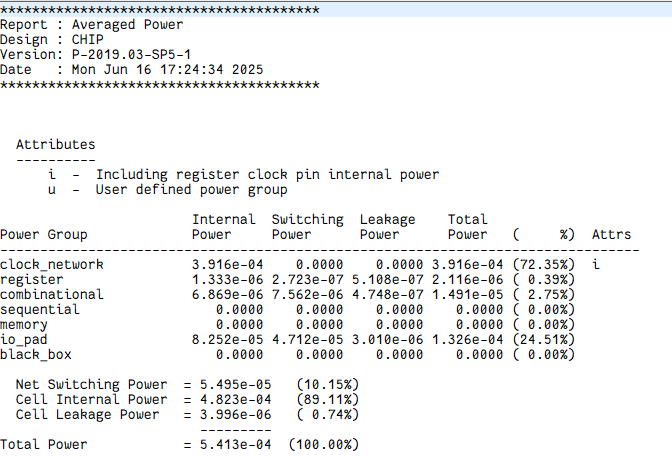
根據pattern 1的toggle情形之power計算:



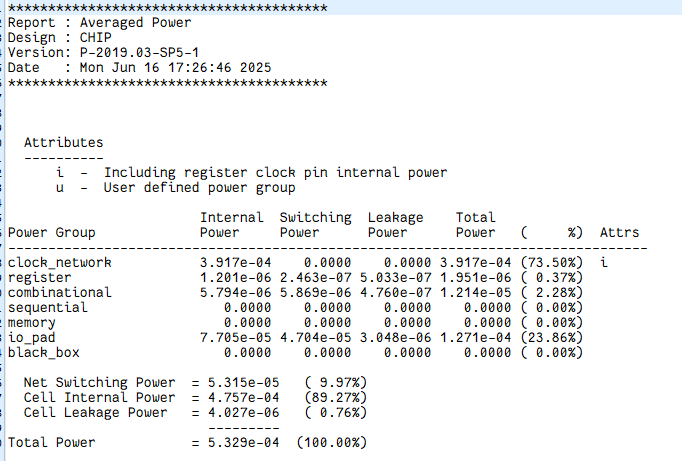
根據pattern 2的toggle情形之power計算:



根據pattern 3的toggle情形之power計算:

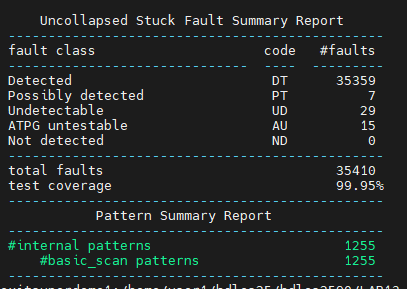


根據pattern 4的toggle情形之power計算:



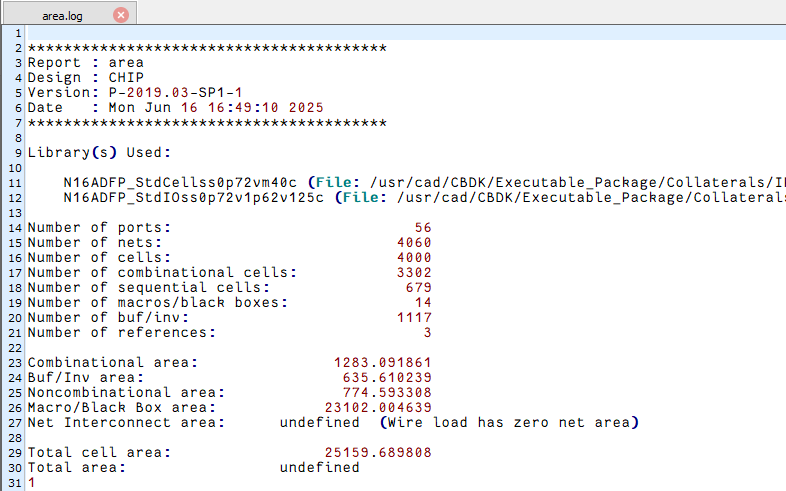
1. DFT驗證與分析 :

DFT Coverage

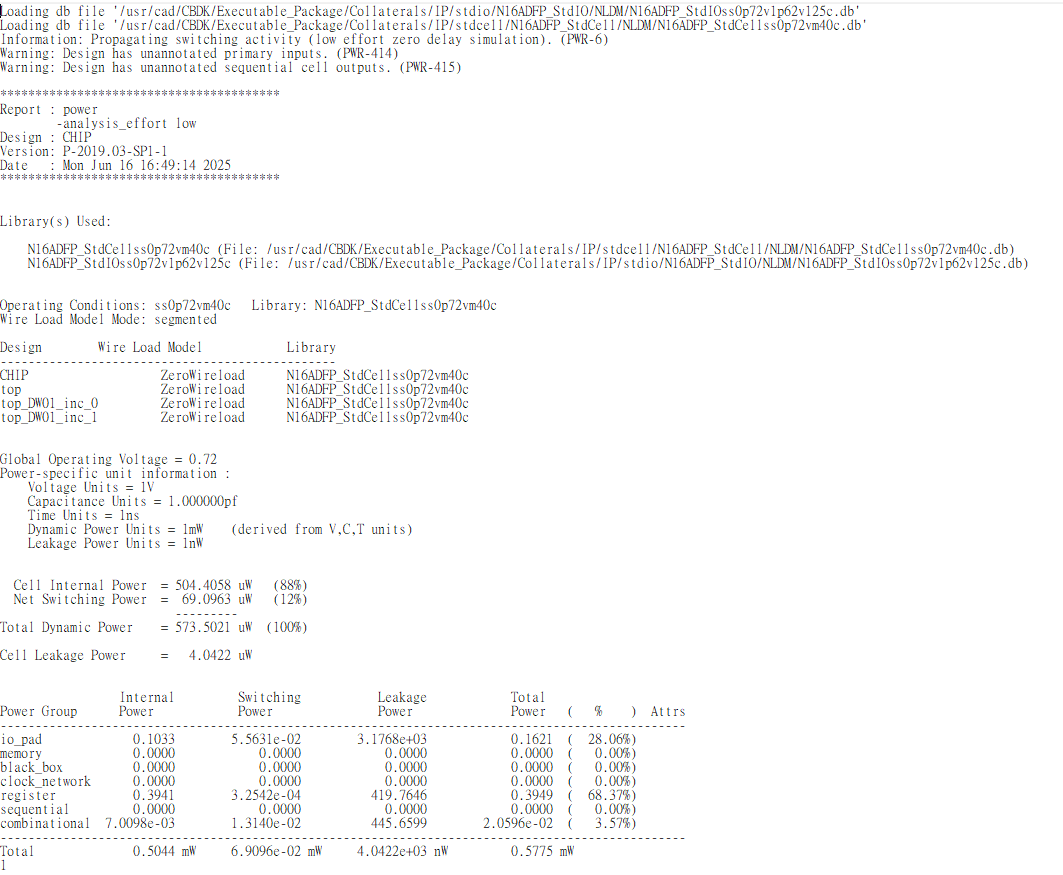


Area/Power before inserting DFT

Area:

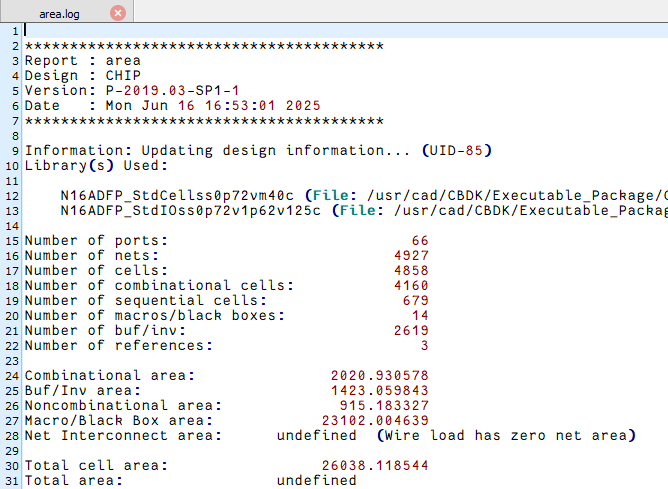


Power:

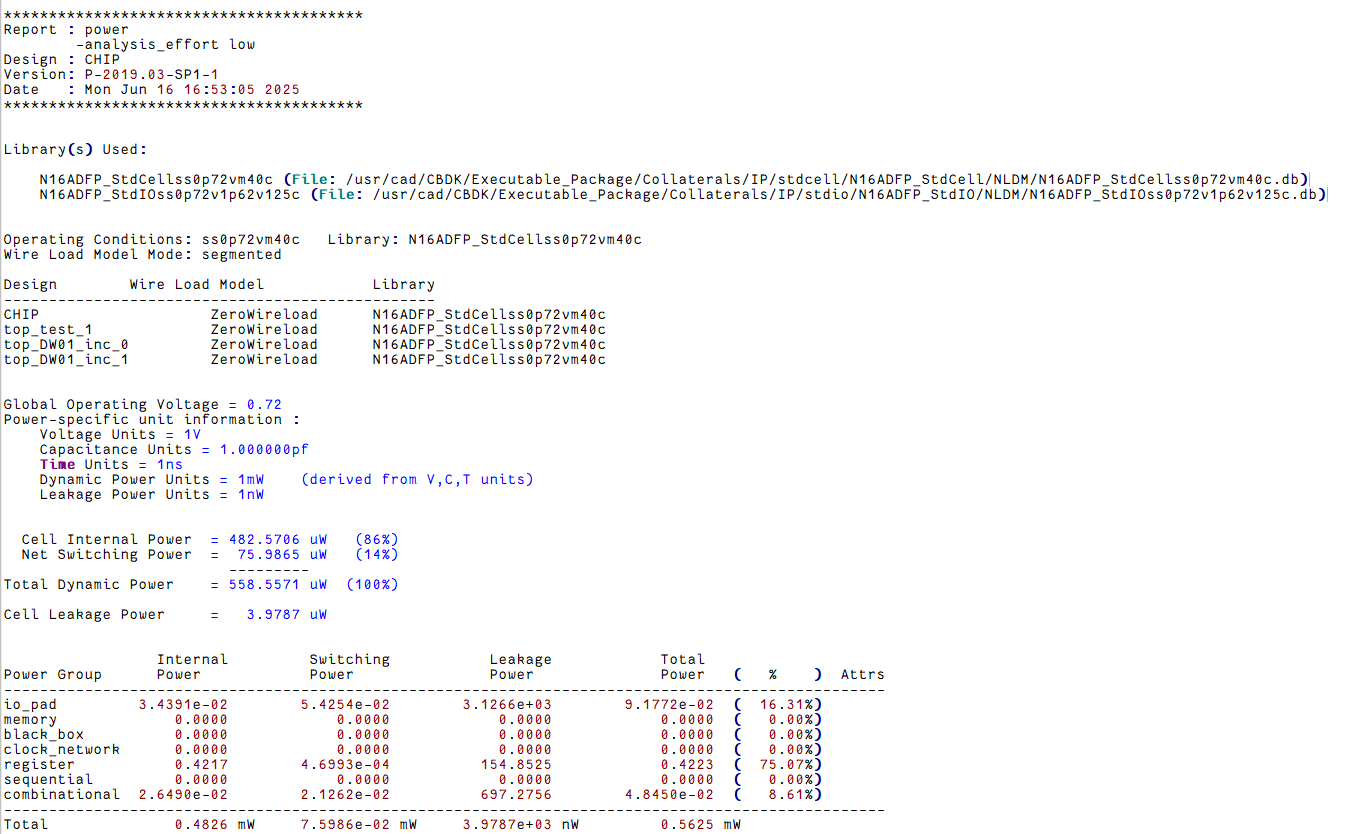


Area/Power after inserting DFT

Area:



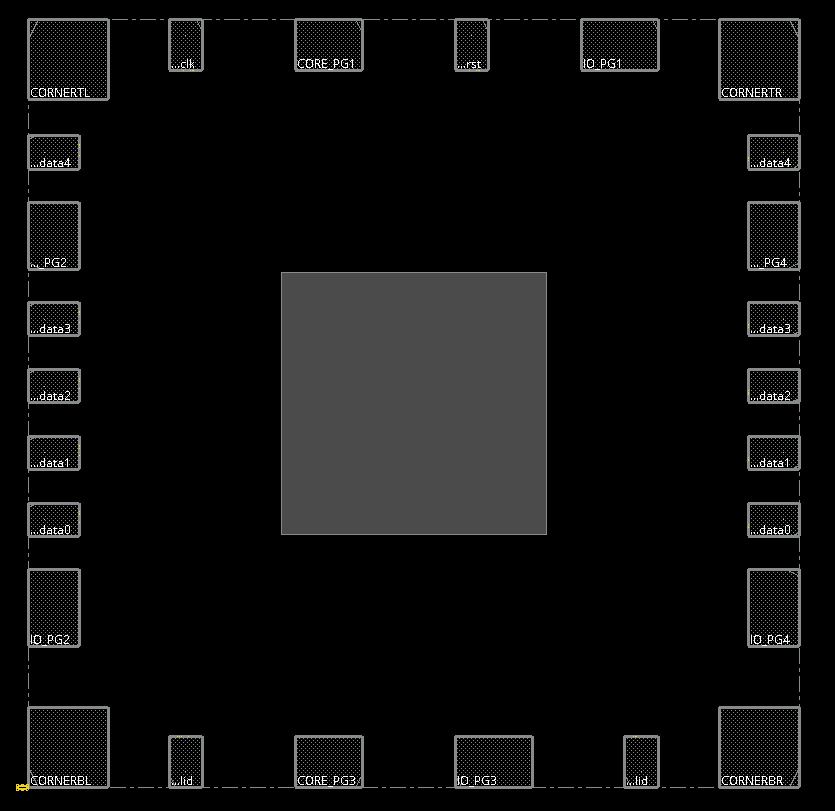
Power:

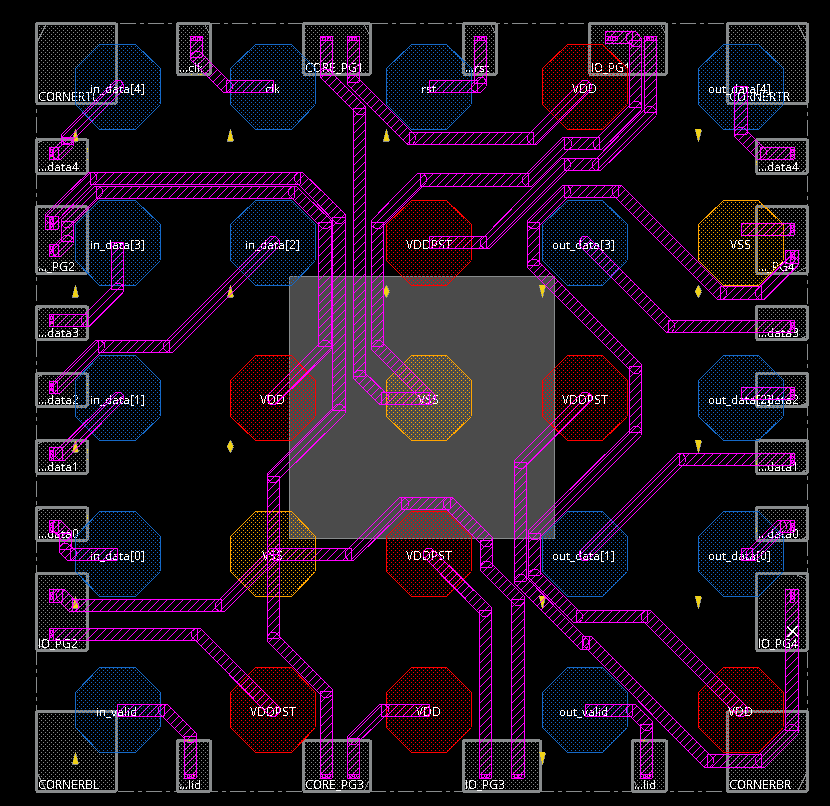


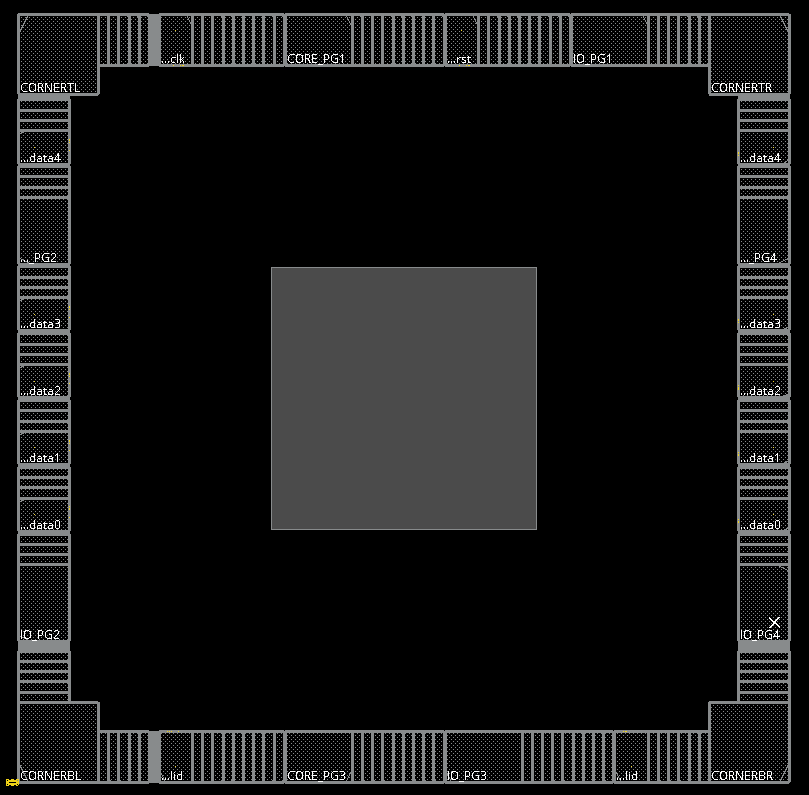
1. APR實作演練與LVS成果 :

Floorplan

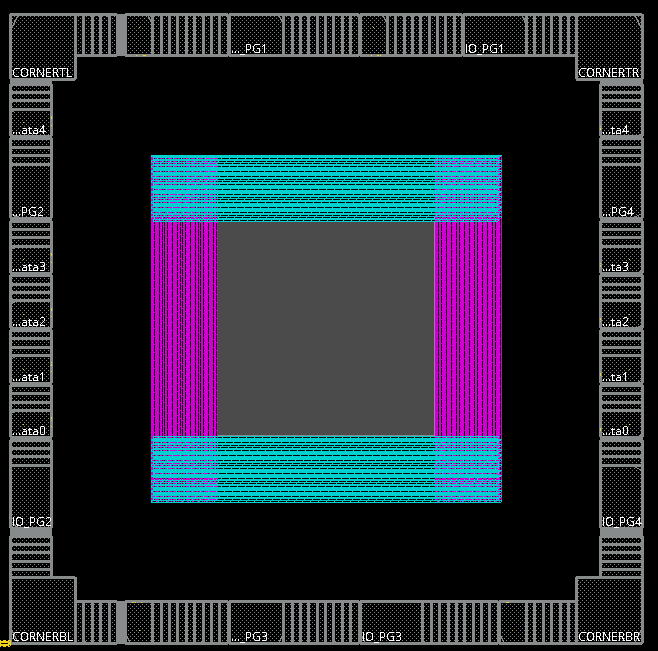
要先擺IO然後對齊FinGrid，並且先看看這樣的擺法以及晶片大小IO能不能藉由bump出去



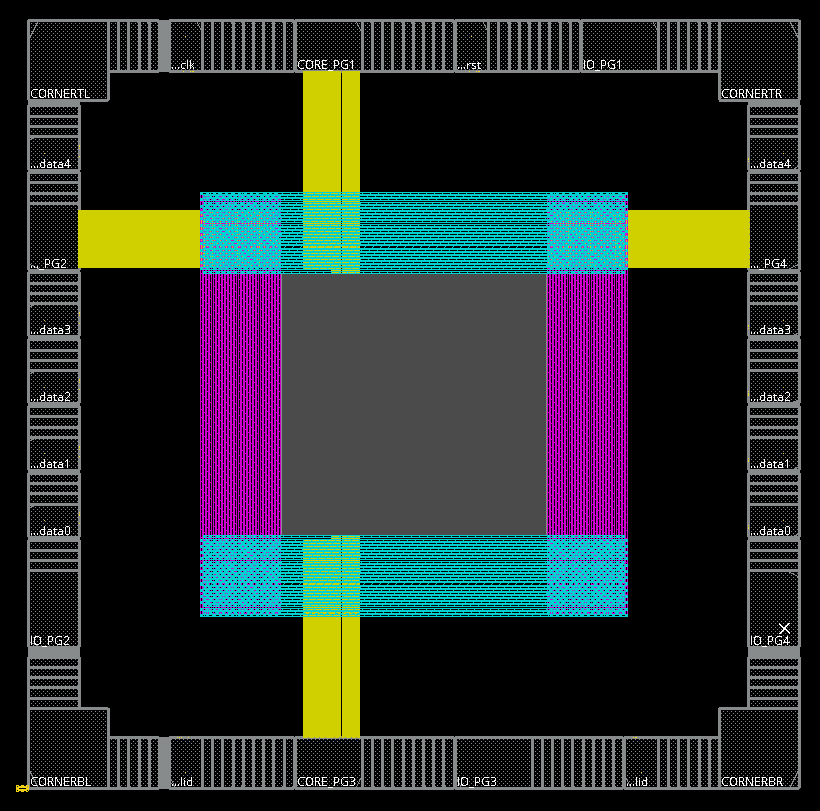
有確認flip chip可以擺上去並且繞線可以RDL繞線成功，原本我的版本為了求過，所以把晶片面積條大到6\*6的bumps才可以繞成功，但助教的script擺得很好，所以5\*5的bumps就可以成功繞出來，底下為RDL繞線成功結果圖  
  
確認可以後再移除，並且最後加上IO filler，以滿足實體上能夠製造晶片的規則



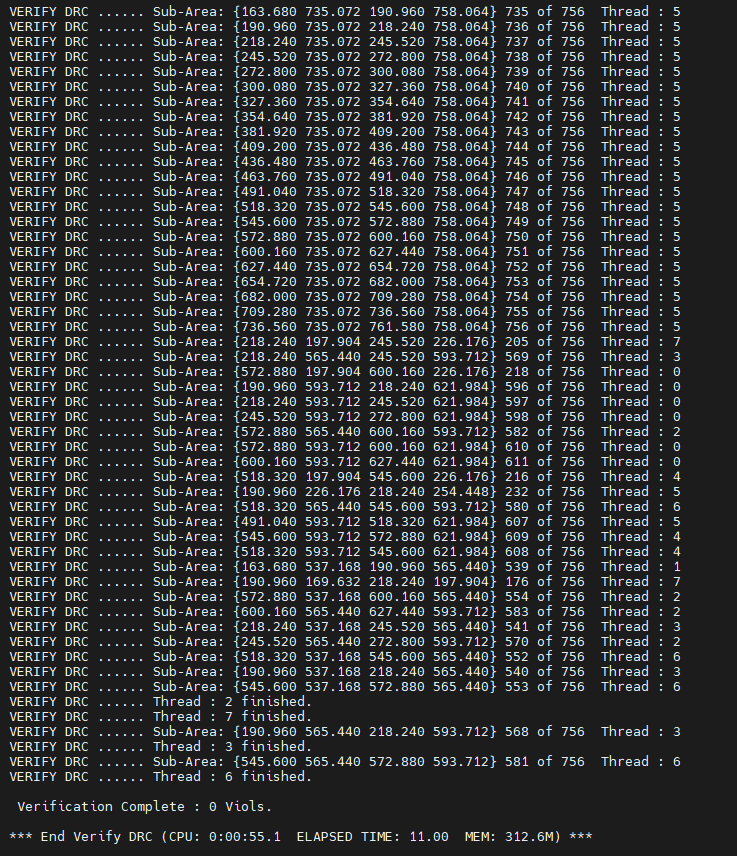
之後是開始進行powerplan，先從power ring開始，為了降低IR，所以會打很多層metal以降低電阻，讓外部電壓進來不會有太多壓降



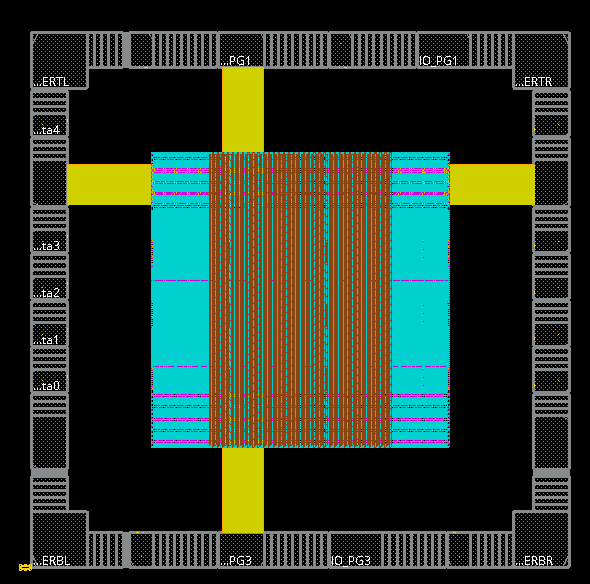
之後是power pad，主要是讓剛剛的power ring接上IO pad來讓他接到外部電源



之後要初步確認到目前步驟下DRC沒有問題

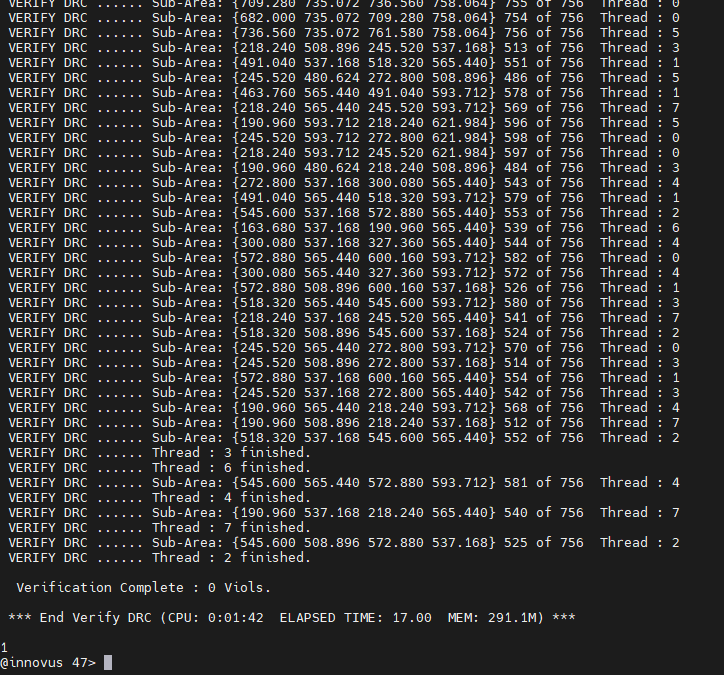


接著是powerstripe，這部是讓剛剛從外部接到powerring的店員在給晶片內部



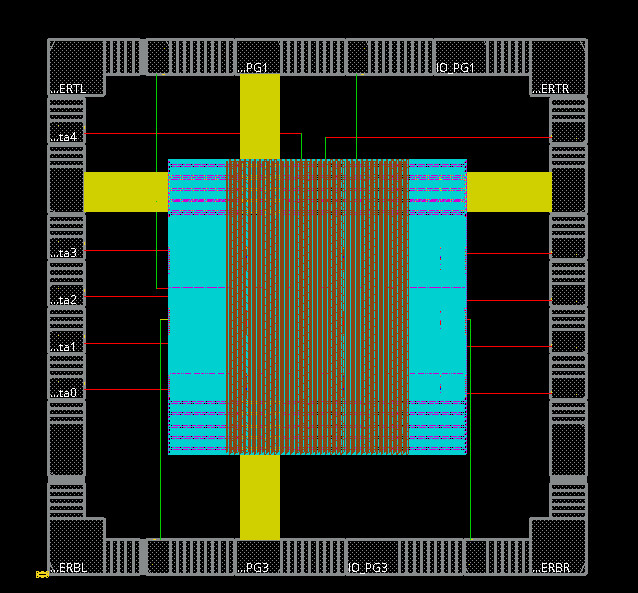
這個之後要經過一連串修DRC的動作，但都助教給好的script，

我放結果



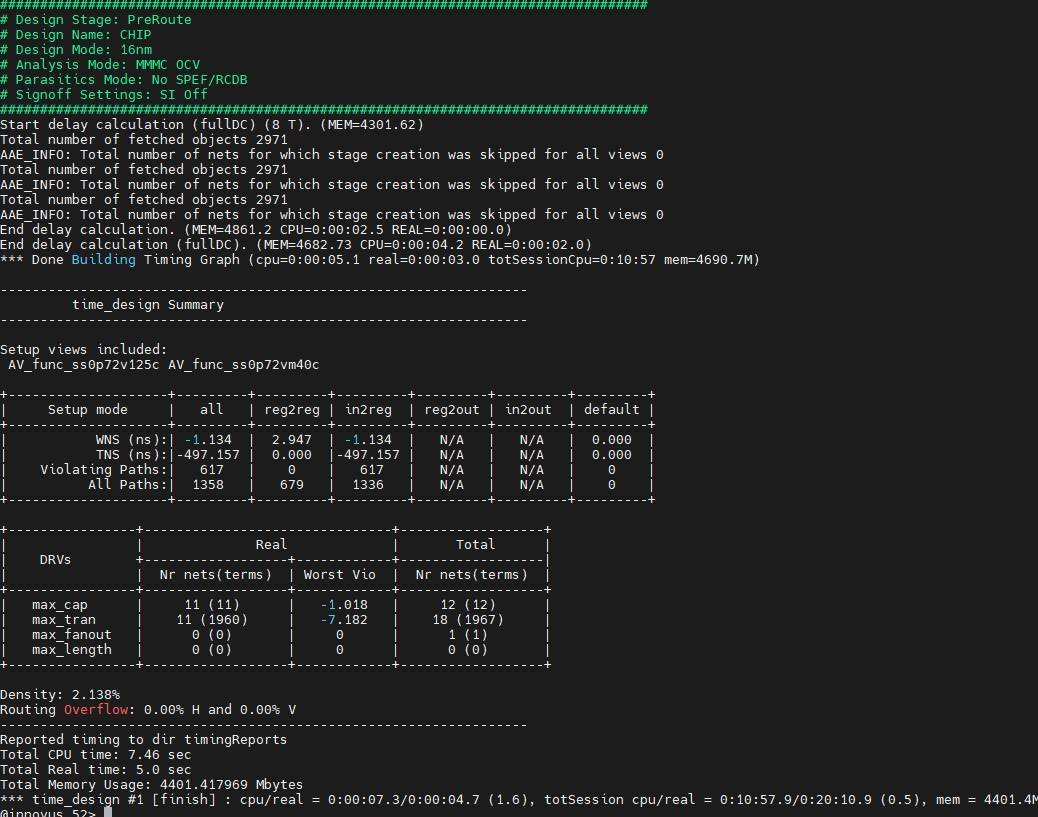
Placement

Placement第一步是加endcap cell，避免切割動作破壞晶片本身，之後該使擺cell，底下維結果

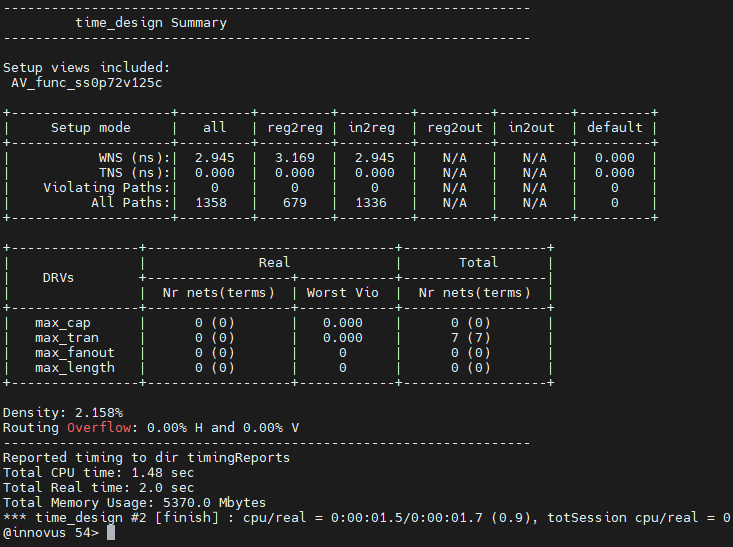


Timing Report Before CTS

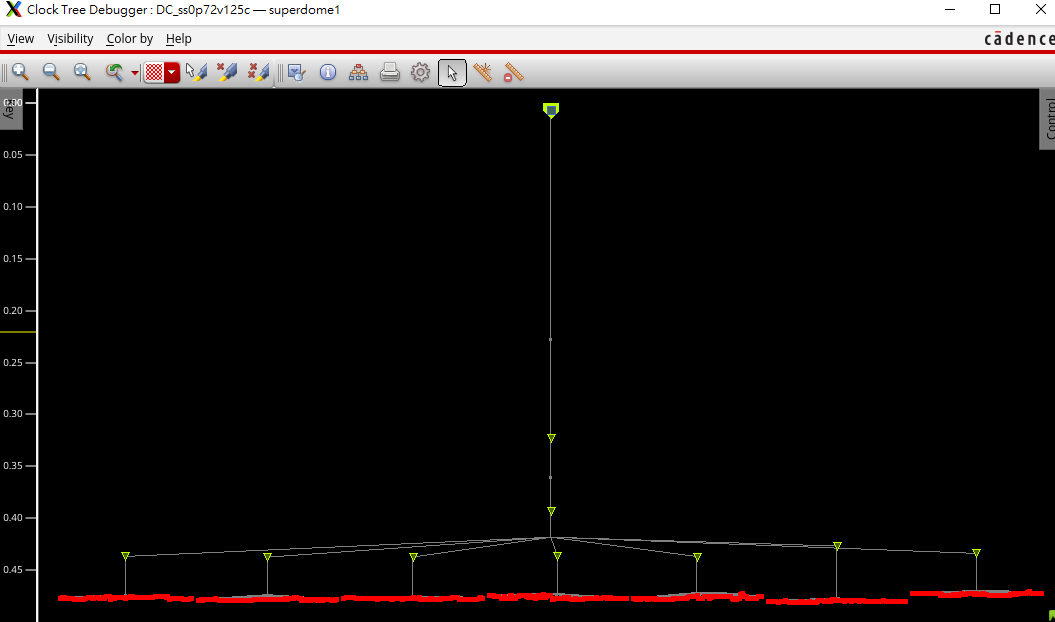
一開始擺時place完，timing沒meet

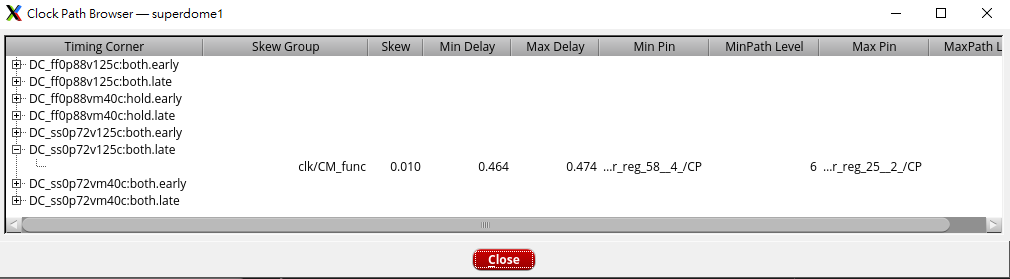


之後下指令place\_opt\_design，來修timing



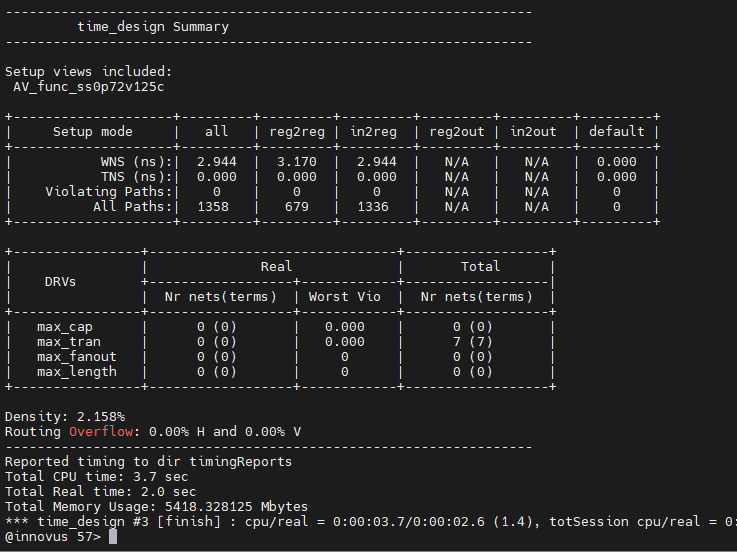
CTS



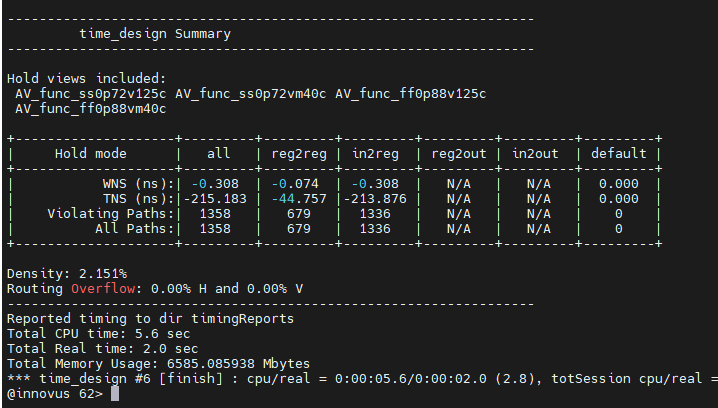


Timing Report After CTS

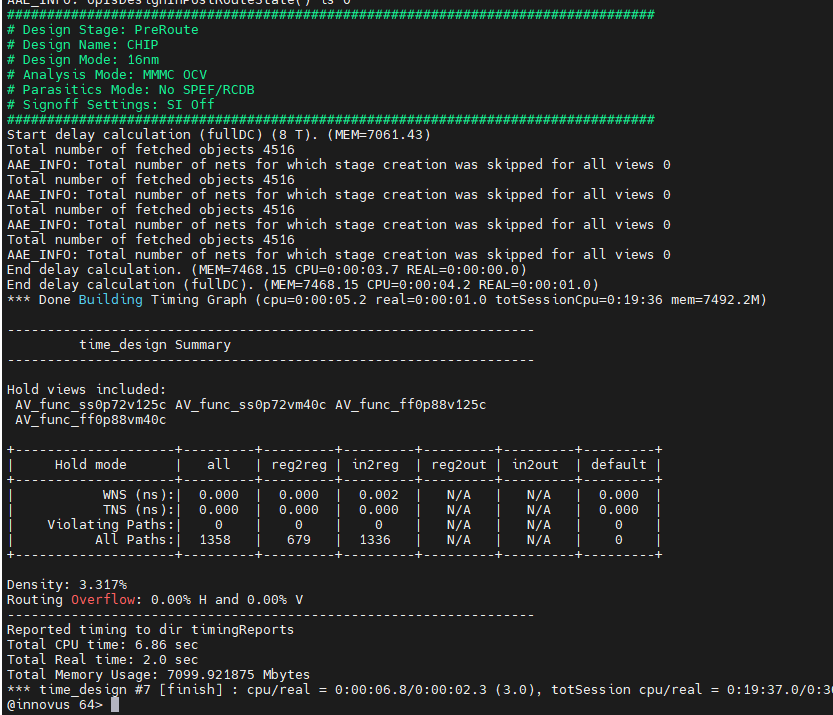
Setup time:



Hold time:

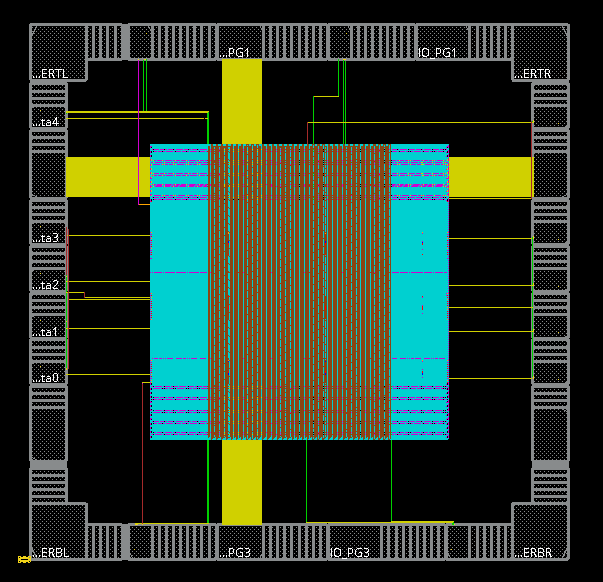


一開始沒過後來下指令opt\_design -post\_cts -hold

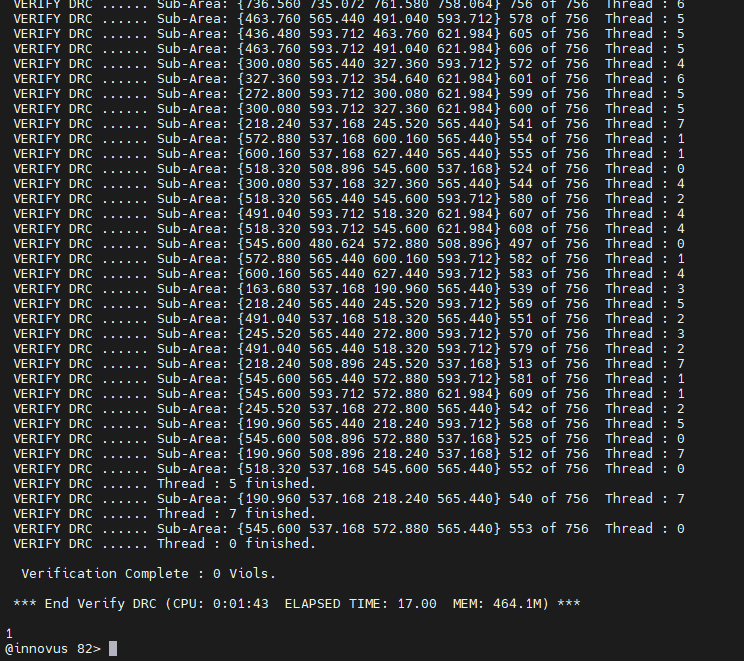


Routing

下指令route design，來繞線，並且要確認timing，底下為結果

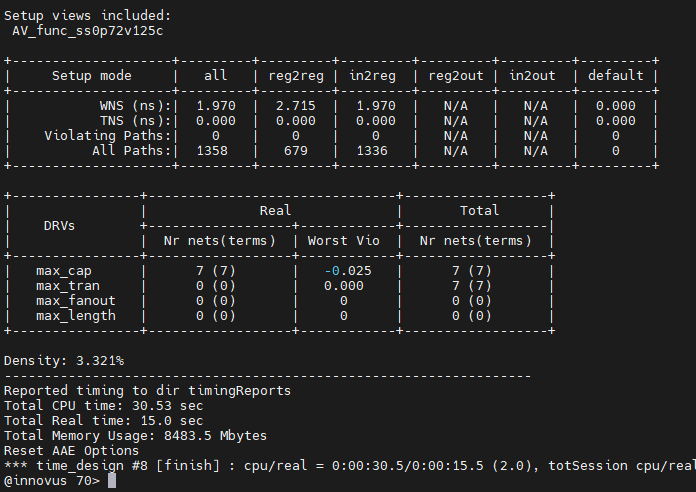


之後就是下一堆指令修DRC，結果有成功修好

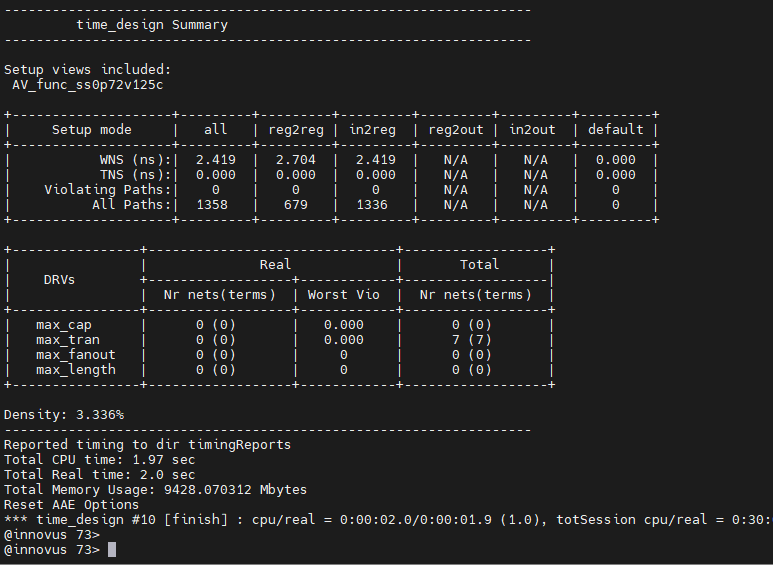


Timing Report After Routing

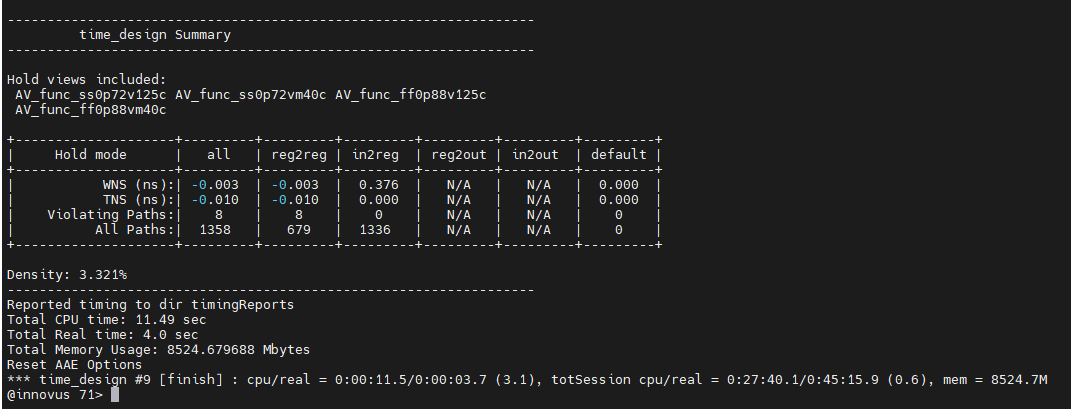
Set up time:



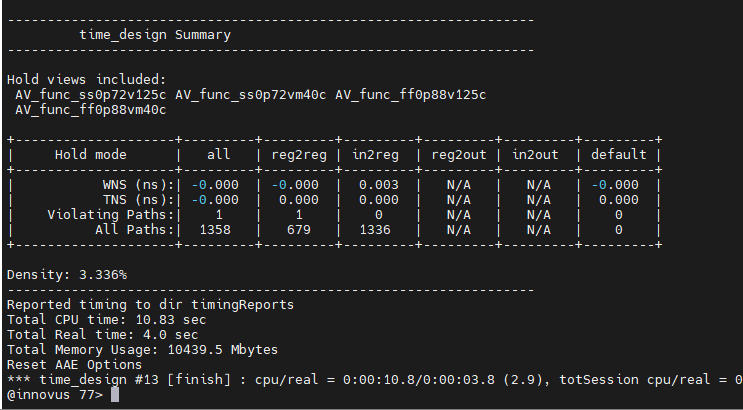
一開始沒過後來下指令，opt\_design -post\_route -setup -hold，就把set up/hold time都修好



Hold time:



一開始沒過後來下指令，opt\_design -post\_route -setup -hold，就把set up/hold time都修好

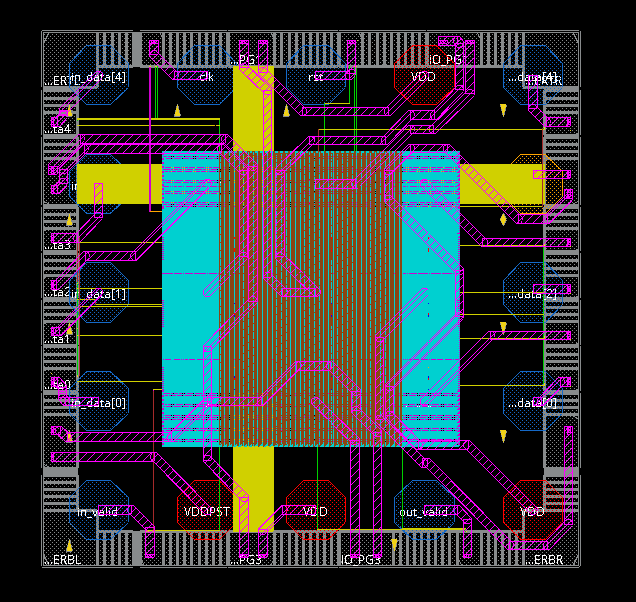


Create Bump

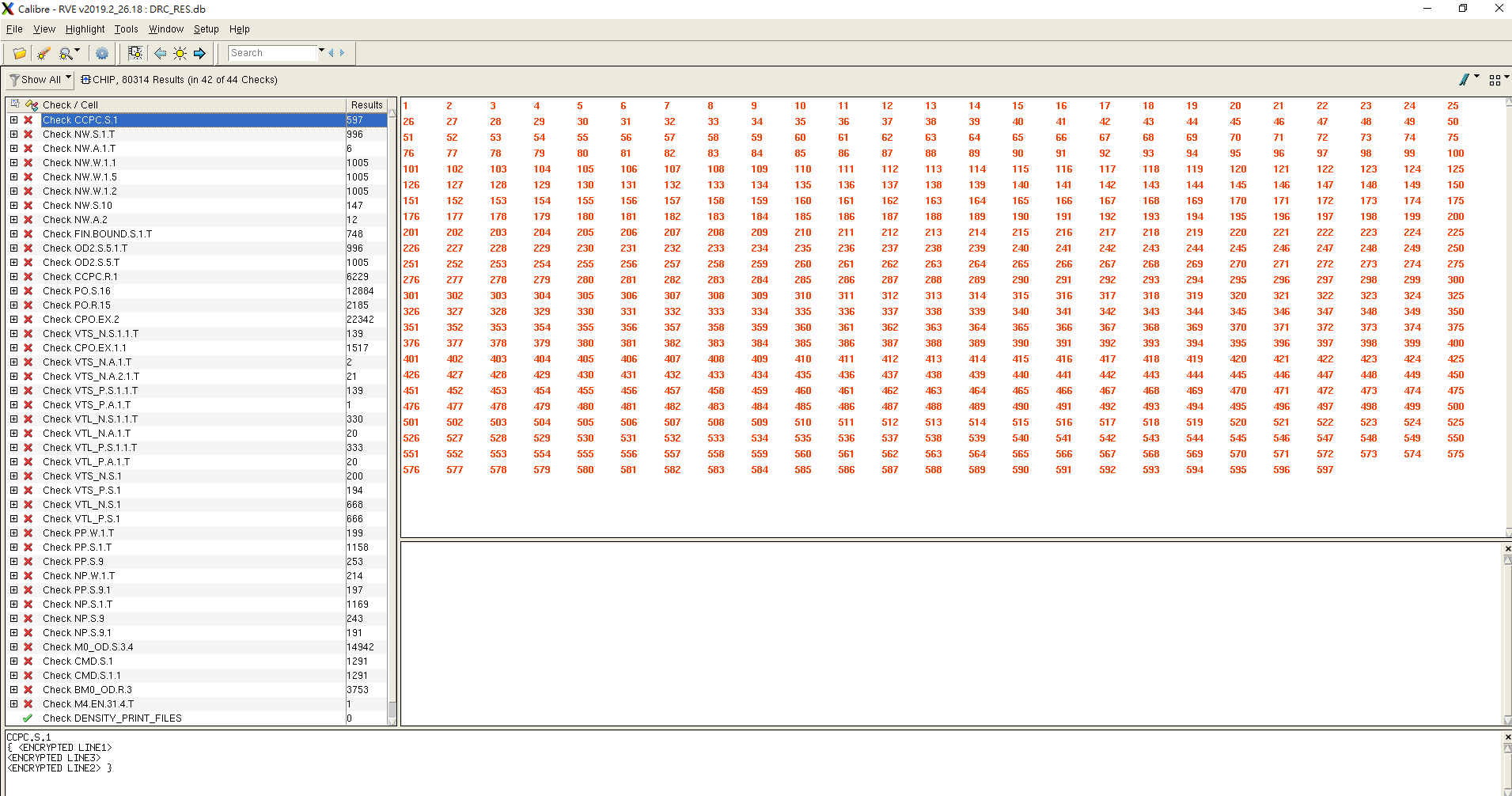
之後就照前面floorplan試擺過的bump擺法再擺回去

RDL Routing

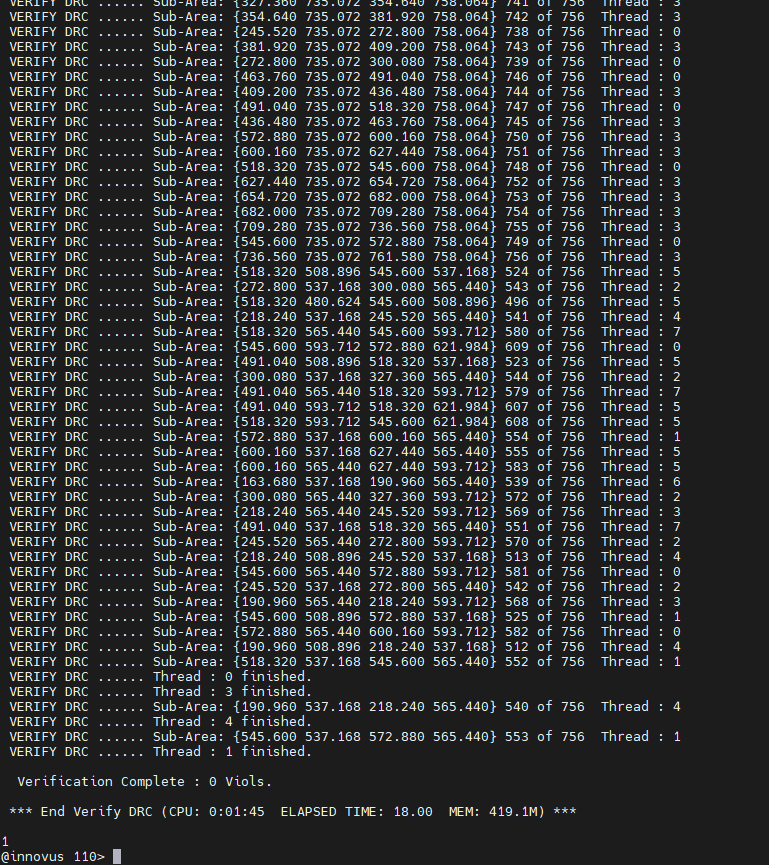
之後就照前面floorplan試繞過的RDL繞法讓tool再繞一次



DRC Check

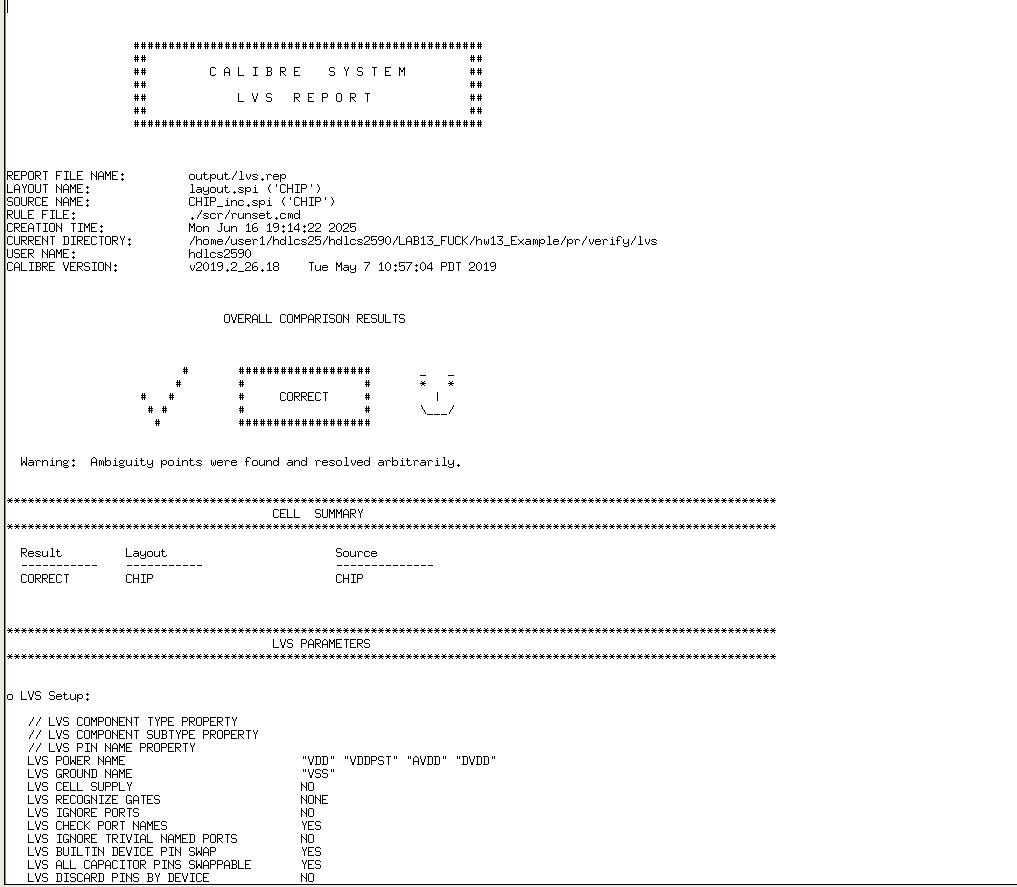


錯一狗票不知為啥，但剛剛繞完再innovous明明是0個，但開calibre出現一狗票，底下是innovous上的



LVS Check

LVS出現了經典笑臉



1. **心得分享**