

VLSI超大積體電路實習

題目:前瞻進位加法器

任課教師：李文達老師

班級：電子三甲

學生：109360142邱少譽

**實習名稱:四位元前瞻式加法器(4Bits Look ahead adder)**

班級: 電子三甲 姓名: 邱少譽 學號:109360142

1. **基本理論:前瞻式加法器(Look ahead adder)**

由於在一般加法器下所生成的+ +

所以我就先行將其公式整理為+

並設兩個變數 來代表 、

可得+ ，並藉由此架構，先行計算出進位的數值

1. **VLSI電路架構設計**
2. 本電路總元件介紹

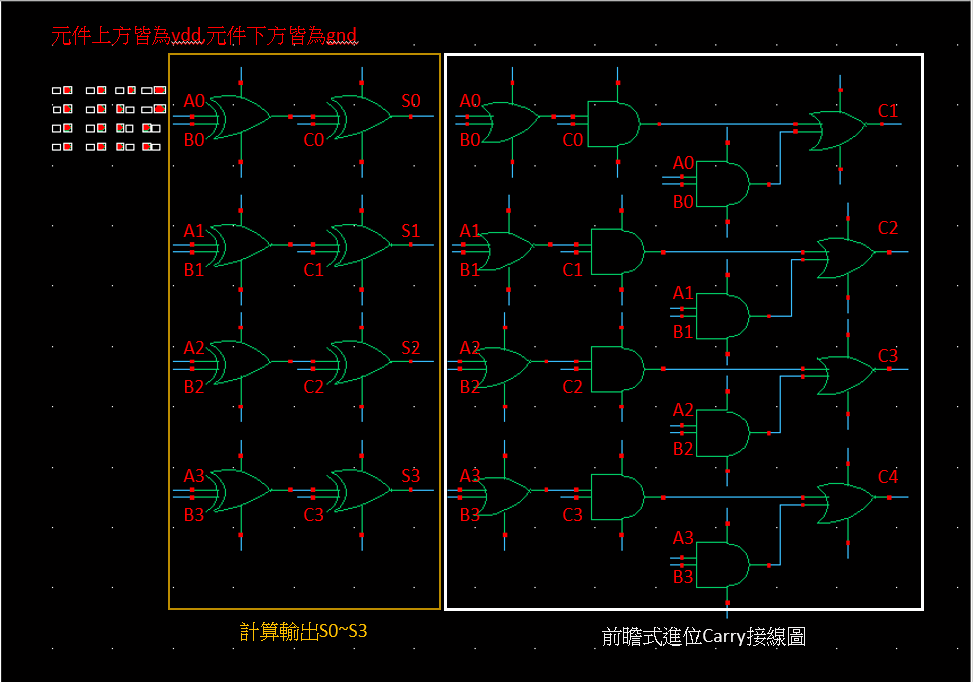
****

圖4:前瞻式加法器cadence電路總圖

(2)本電路的製成步驟如下:

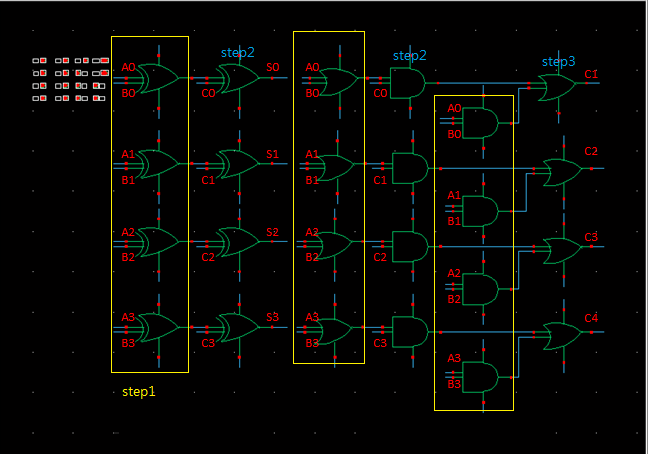


圖1:輸出電路步驟架構圖

STEP1:先計算An ⊕ Bn(左) , An or Bn(中) , An and Bn(右)

STEP2:接續計算An ⊕ Bn ⊕ Cn 得Sn，以及

STEP3接續計算得+

STEP4:重複Step2 與 Step3，即可得解

(3) 提前計算Cout的電路方式

可直接算出Cout並且能夠多節省兩個時間序，但是在需多接多個多輸入的邏輯閘，所以本電路並未採用，藉由上述+ 可以更快的一層層推敲從C1到C3，特別的是在最高為能先算carry，所以能夠多節省兩個時間序。

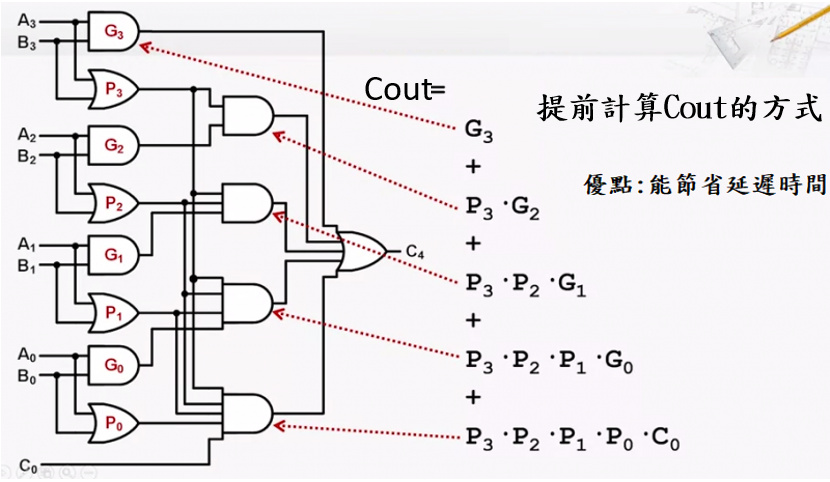
****

圖2:提前計算Cout的電路架構(圖片來源:it邦幫忙)

(4)與其他比較器比較優缺點

表1:漣波加法器(RCA)與前瞻加法器(CLA)比較

|  |  |  |  |
| --- | --- | --- | --- |
|  | 漣波加法器(RCA) | 前瞻加法器(CLA) | 前瞻加法器 (先算Cout) |
| 組成結構 | 將四個一位元加法器串接而成 | 每個位元的進位是來自前瞻加法器所計算出的結果 | C1到C3的進位是來自前瞻加法器所計算出的結果 |
| 優點 | 電路較簡單，設計方便 | 進位延遲時間能夠個別先算，較節省時間 | 進位延遲時間能夠個別先算，且最後Cout可以先算能更省時間 |
| 缺點 | 高位元運算需要等低位元運算的進位，所以延遲時間較長 | 若位元數一多，電路較為複雜，較難設計 | 若位元數一多，電路比一般前瞻式加法器更為複雜，更難設計 |
| 花費時間比較  (以nbits為例)(如圖3) | 3n | 1+2n | 1+2n-2 |

(4)解析所延遲的時間

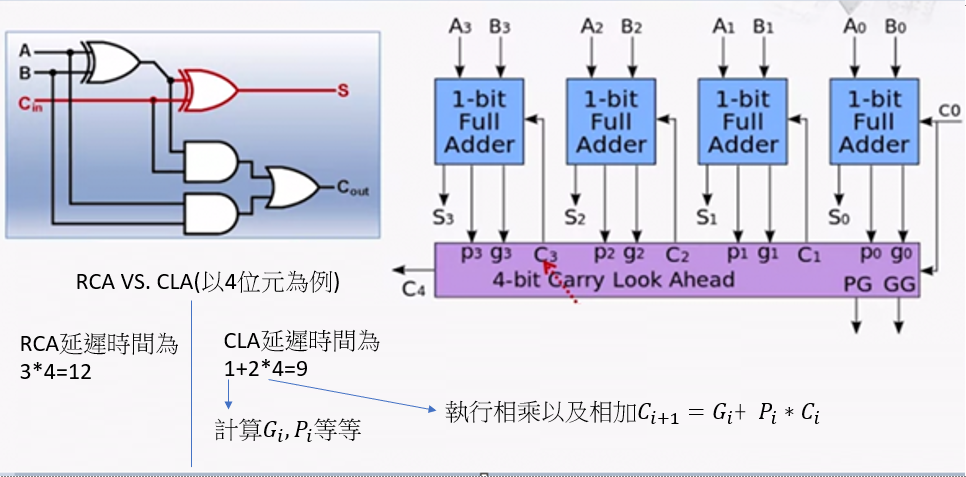
****

圖3:漣波加法器與前瞻加法器延遲時間及架構解析(圖片來源:it邦幫忙)

1. **元件佈局(SP檔展示)**

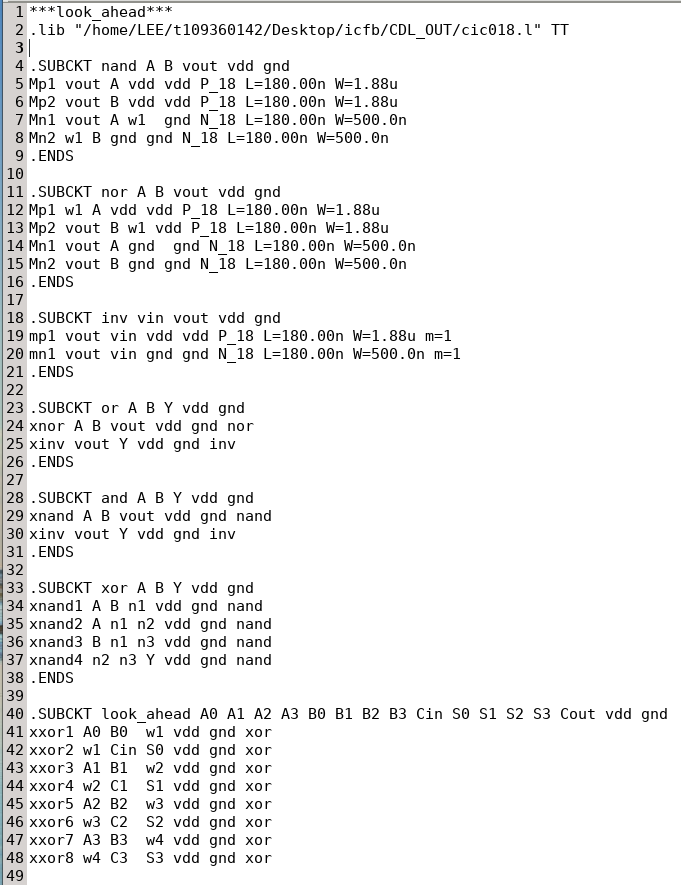
****

圖5:sp檔1

SP檔功能介紹:

(如圖5)第二行:引用cic018.l製程檔(TT為模擬環境)

(如圖5).subckt到.ends為各個包裝好的子電路

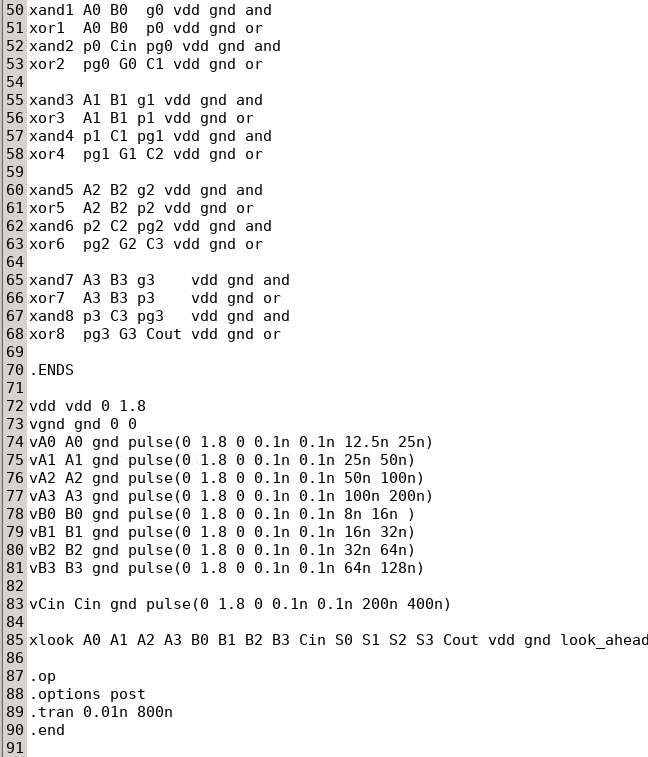
****

圖6:sp檔2

SP檔功能介紹:

(如圖6)第74行的pulse function 內部數值為:

PULSE (V1 V2 td tr tf pw per) m+ Y: g! M1 ~6 [8 O  
V1  初始電壓,V2  脈波電壓td,延遲時間,tr  上升時間

,tf  下降時間,pw  脈波寬度,per 週期

數值設計理念per為pw的兩倍，且每個位元差的per也同為兩倍就能滿足所有解，(A與B的per時間不同能夠錯開，以方便測試)

1. **晶片電路波形模擬**

以下波形第一個為cin,第二個為A3~A0,第二個為B3~B0第四個為cout第五個為S3~S0，由圖8可得大致模擬的結果

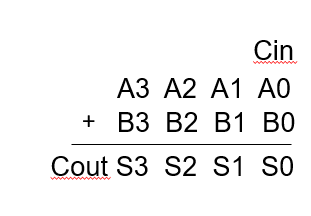
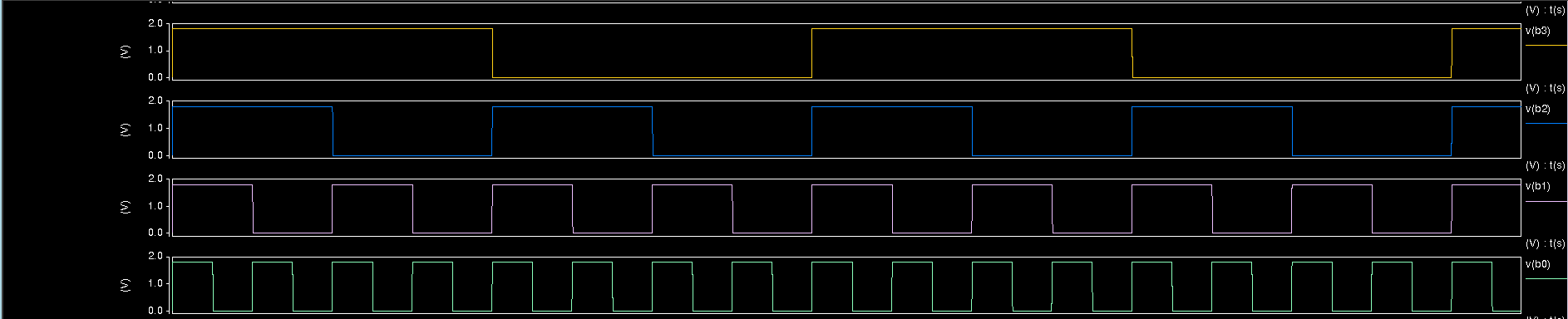
****

圖7:輸出算式對照圖

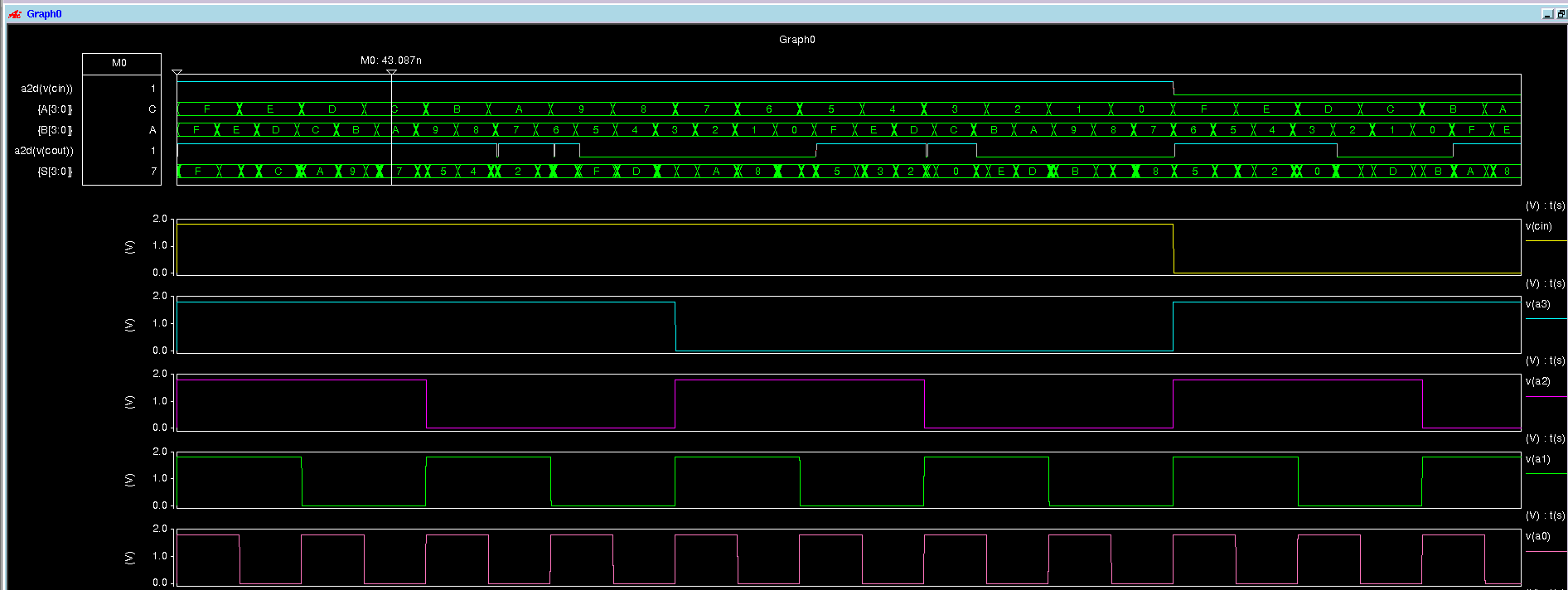
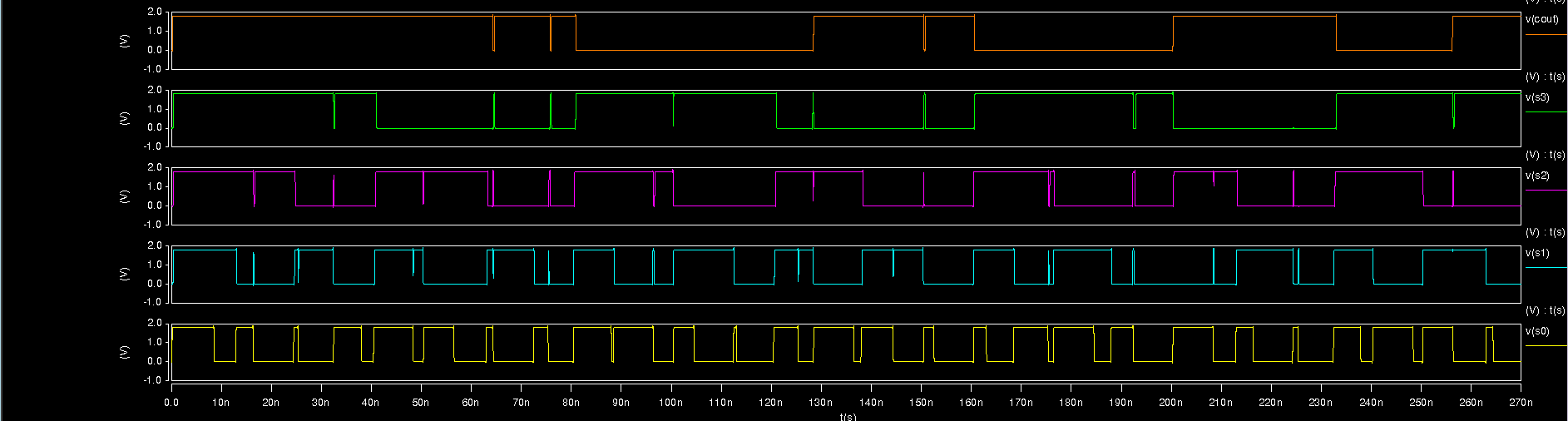
****

圖8:前瞻式加法器cosmos scope模擬輸出入腳位

1. **LVS結果**

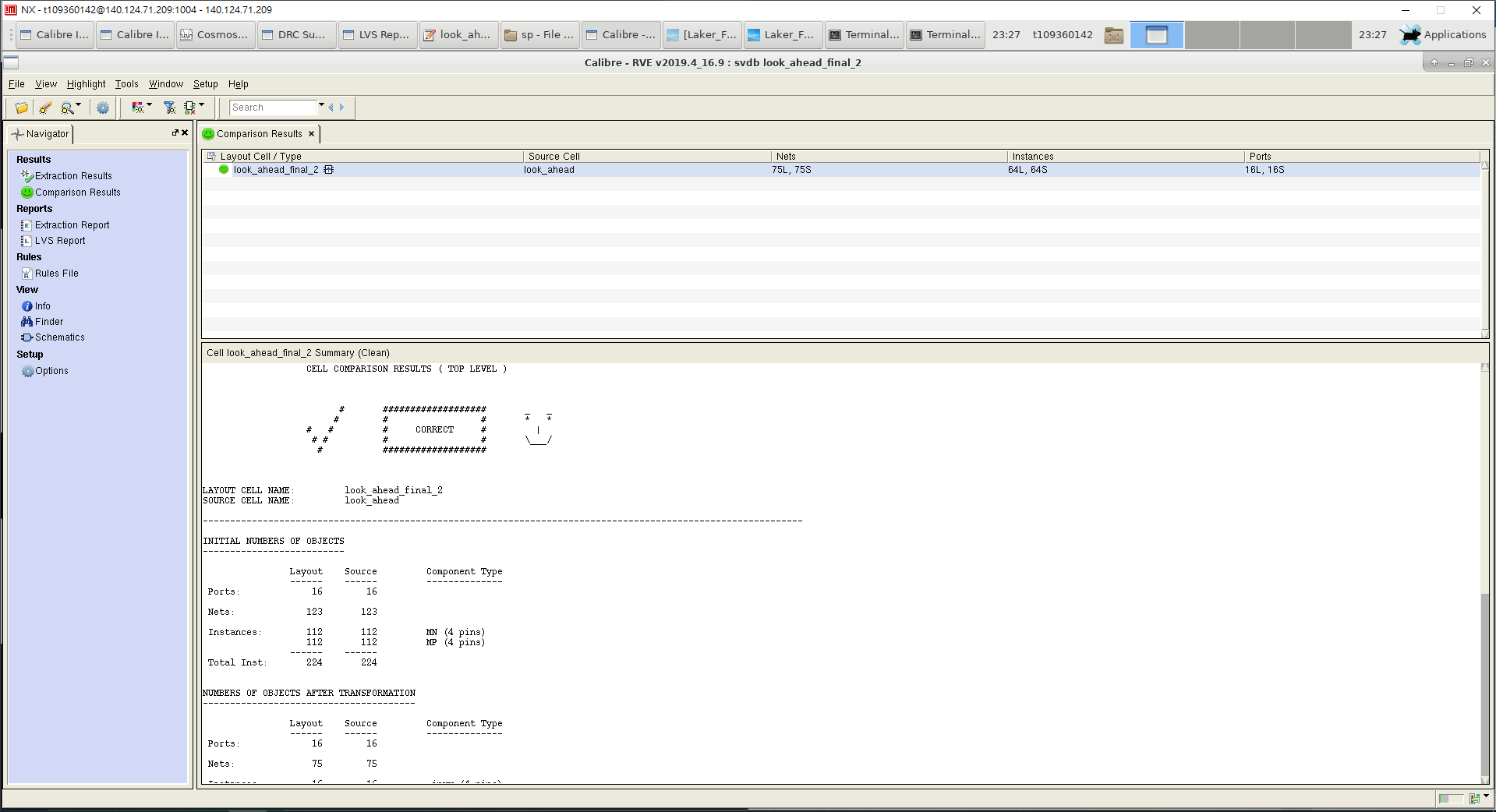
****

圖9:前瞻式加法器LVS

1. **佈局結果**

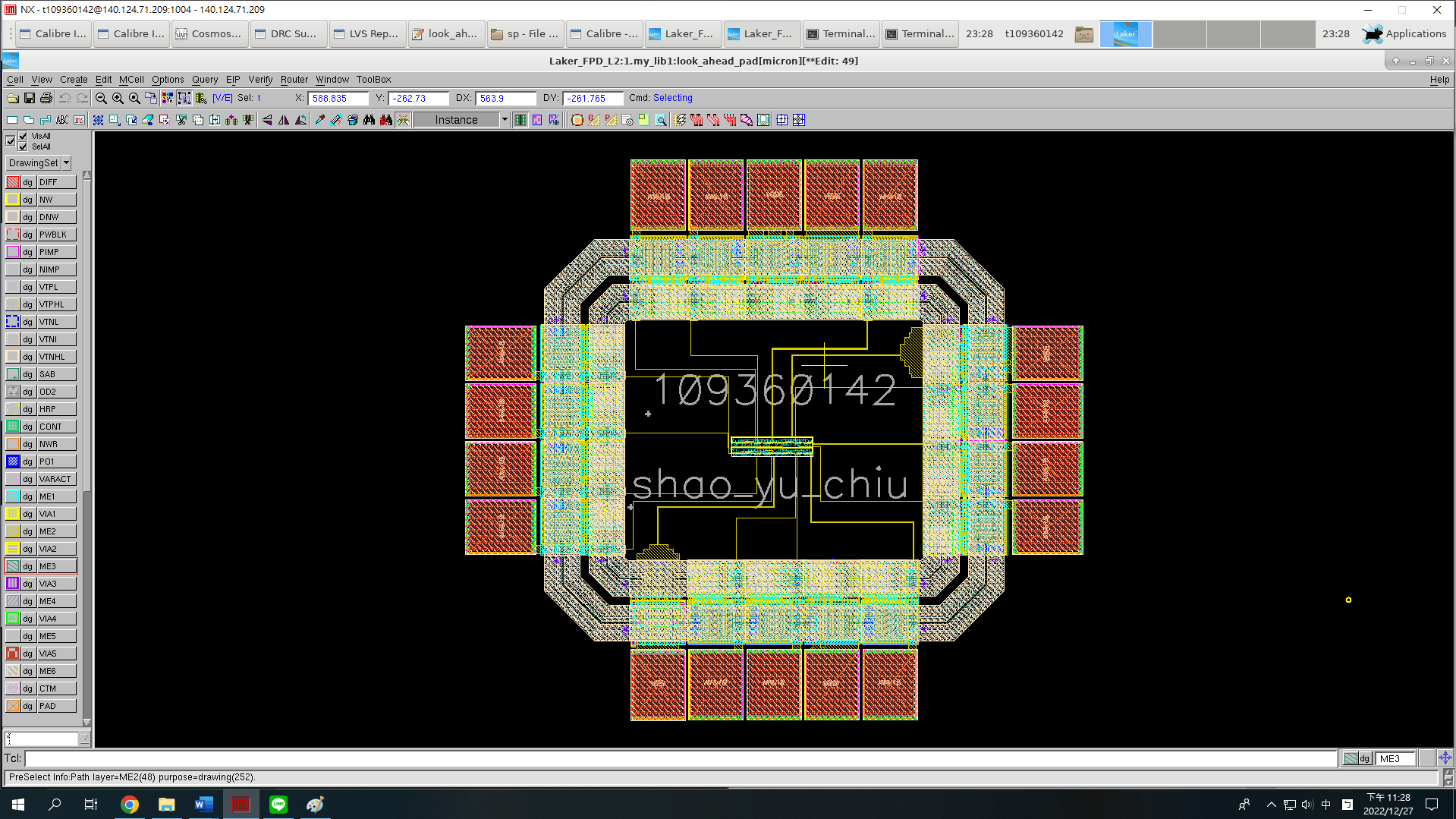
****

圖10:前瞻式加法器整體布局結果(含PAD)

Pad接線腳位簡介:AIN\_18為輸出入的PAD,VDDI為內層電路的電源端,VSSI為內層電路的接地端,VDDE為PAD的電源端,VSSE為PAD的接地端,corner為轉角處的連接端

1. **結論**

藉由本次前瞻進位加法器實驗，有別於以往一般的漣波加法器，雖然架構稍微複雜些許，但是就整體而言相較於省下的時間成本，比起來根本不算什麼，至於本次實驗看似沒有減少很多單位時間，是因為bits數較少，倘若bits數較多時減少的時間就很可觀。