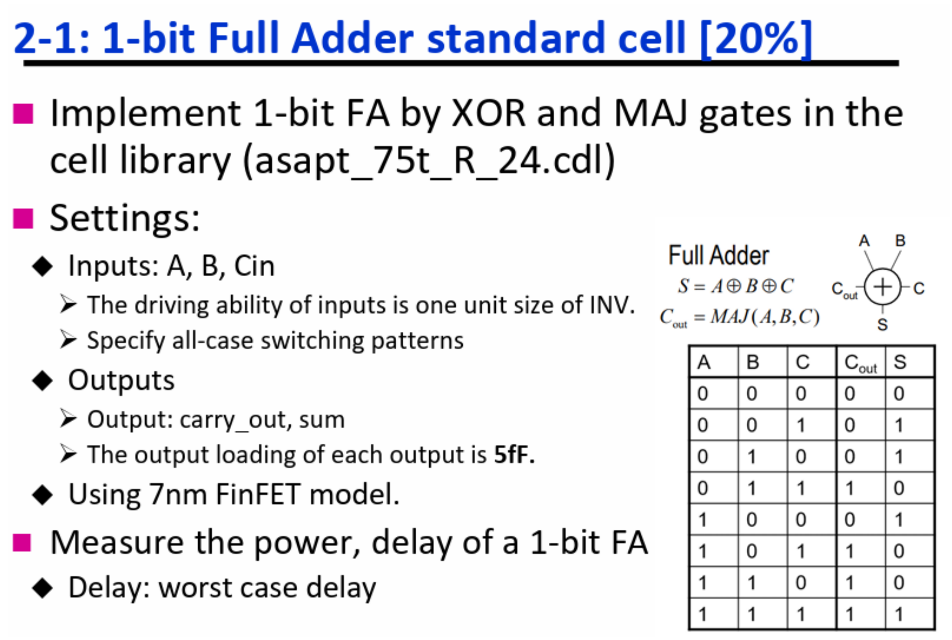
DIC 作業 2 學號: 312510017 宋彥霆



2-1: 1-bit Full Adder standard cell [20%]

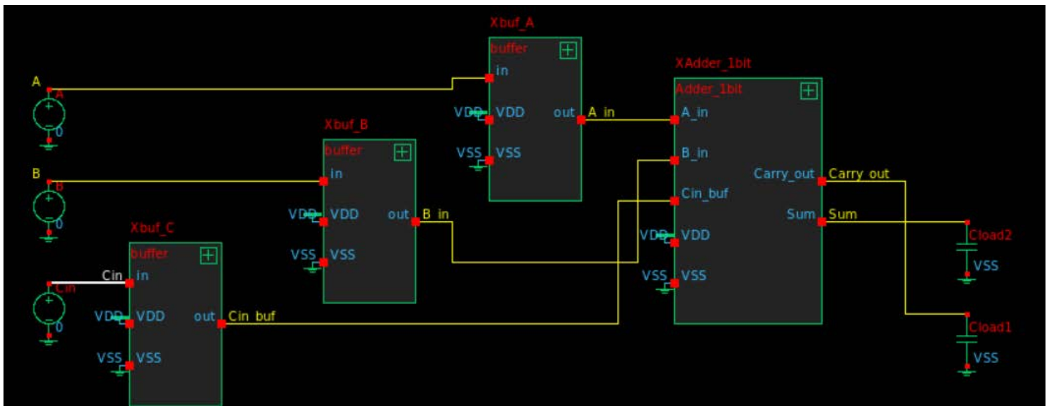
- Implement 1-bit FA by XOR and MAJ gates in the cell library (asapt_75t_R_24.cdl)
- Settings:
  - Inputs: A, B, Cin
    - The driving ability of inputs is one unit size of INV.
    - Specify all-case switching patterns
  - Outputs
    - Output: carry_out, sum
    - The output loading of each output is **5fF.**
  - Using 7nm FinFET model.
- Measure the power, delay of a 1-bit FA
  - Delay: worst case delay

Full Adder

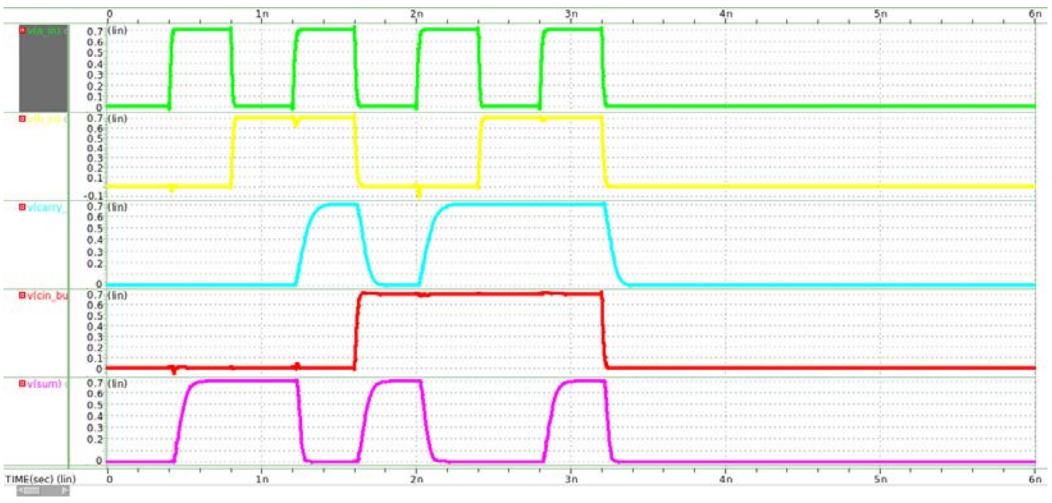
$S = A \oplus B \oplus C$

$C_{out} = MAJ(A,B,C)$

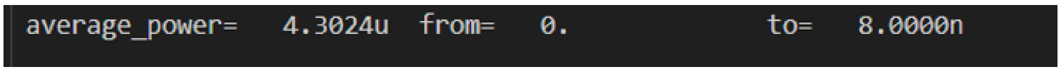
A	B	C	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



以下為以 asapt_75t_R_24.cdl 為 library 做的 one-bit full adder 之實現



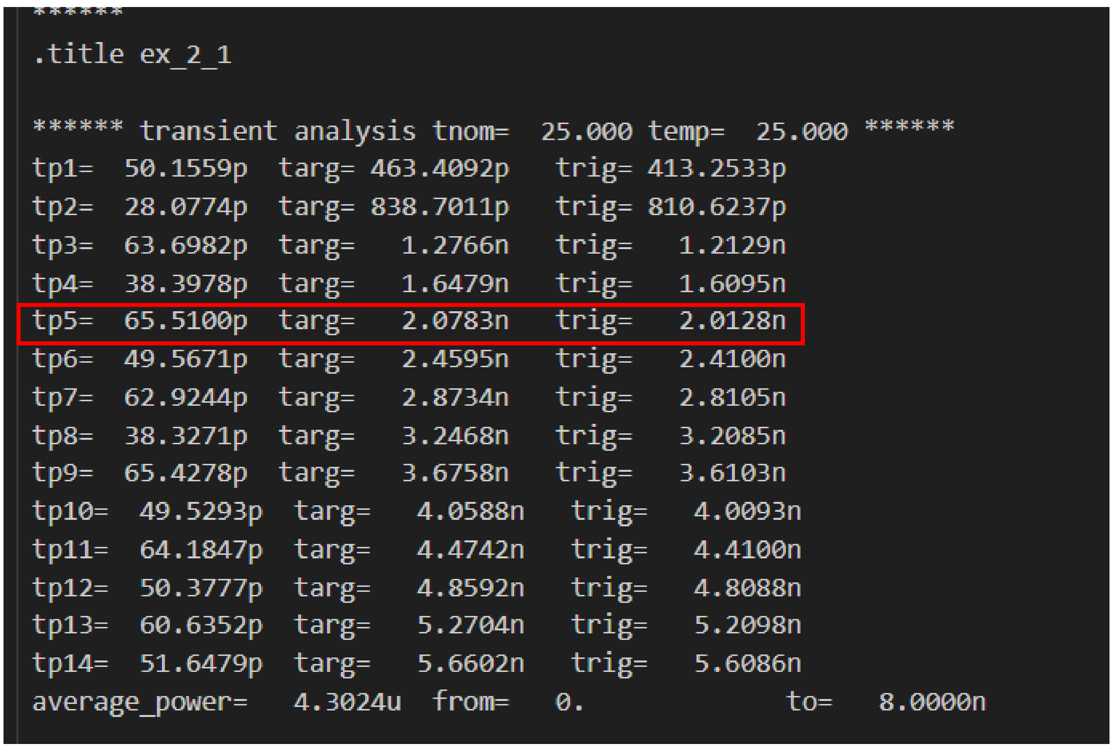
Power:



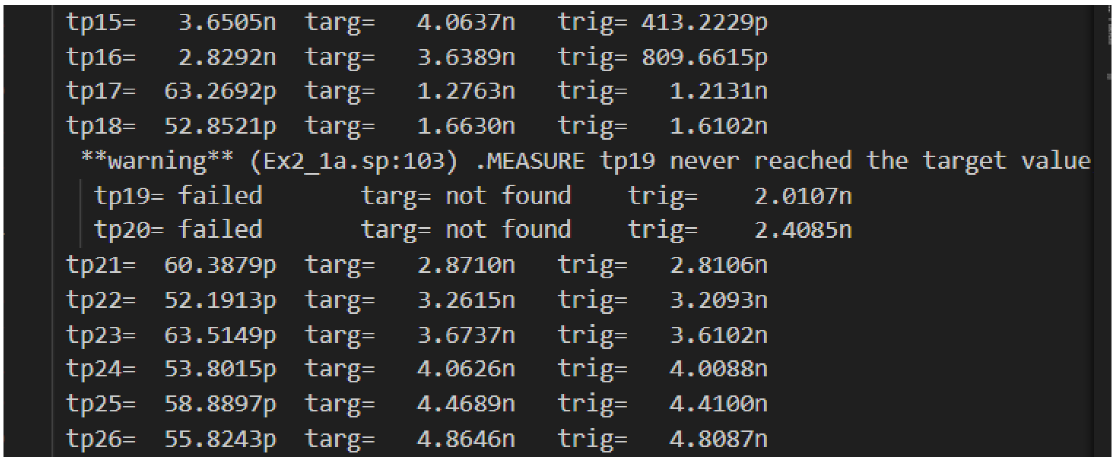
```
average_power=   4.3024u  from=   0.          to=   8.0000n
```

Delay:

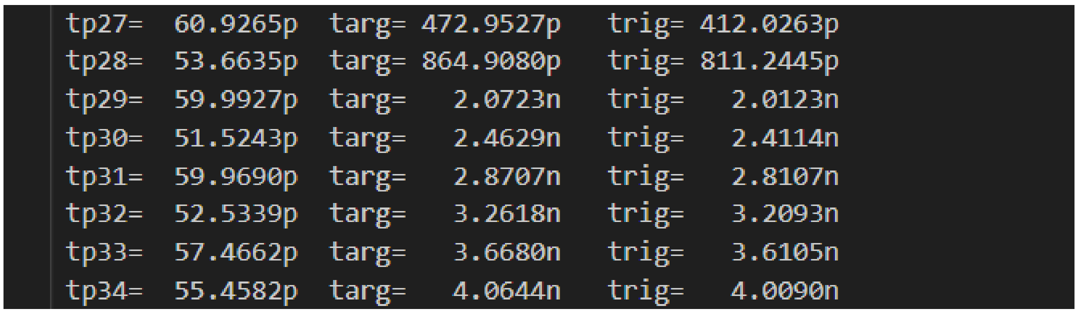
以下考慮了 8*8=64 種排列組合，然後再扣除自己和自己本身 8 種，還有一些 input 變化後 sum 和 carry 訊號沒有變的可能(像是 tp19、tp20 為例，其他就沒量出來，扣除了)後，下去測量知所有結果:

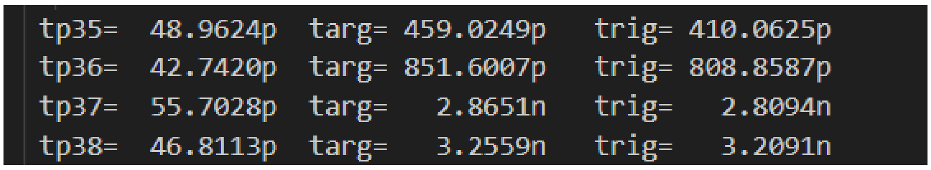


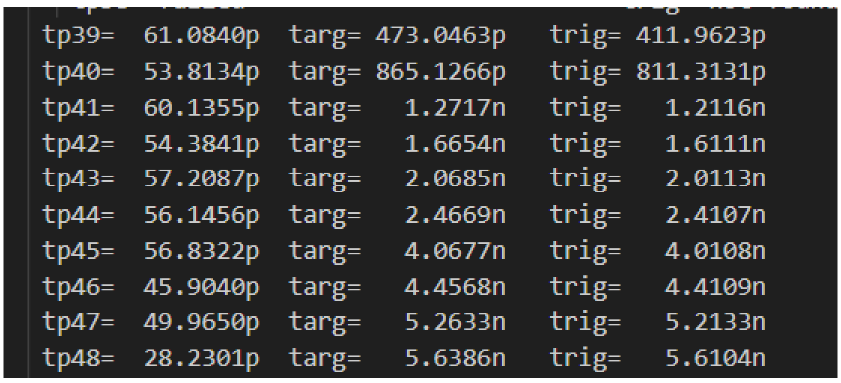
```
 ******
 .title ex_2_1

 ****** transient analysis tnom=  25.000 temp=  25.000 ******
 tp1=  50.1559p  targ= 463.4092p   trig= 413.2533p
 tp2=  28.0774p  targ= 838.7011p   trig= 810.6237p
 tp3=  63.6982p  targ=   1.2766n   trig=   1.2129n
 tp4=  38.3978p  targ=   1.6479n   trig=   1.6095n
 tp5=  65.5100p  targ=   2.0783n   trig=   2.0128n
 tp6=  49.5671p  targ=   2.4595n   trig=   2.4100n
 tp7=  62.9244p  targ=   2.8734n   trig=   2.8105n
 tp8=  38.3271p  targ=   3.2468n   trig=   3.2085n
 tp9=  65.4278p  targ=   3.6758n   trig=   3.6103n
 tp10=  49.5293p  targ=   4.0588n   trig=   4.0093n
 tp11=  64.1847p  targ=   4.4742n   trig=   4.4100n
 tp12=  50.3777p  targ=   4.8592n   trig=   4.8088n
 tp13=  60.6352p  targ=   5.2704n   trig=   5.2098n
 tp14=  51.6479p  targ=   5.6602n   trig=   5.6086n
 average_power=   4.3024u  from=   0.          to=   8.0000n
```



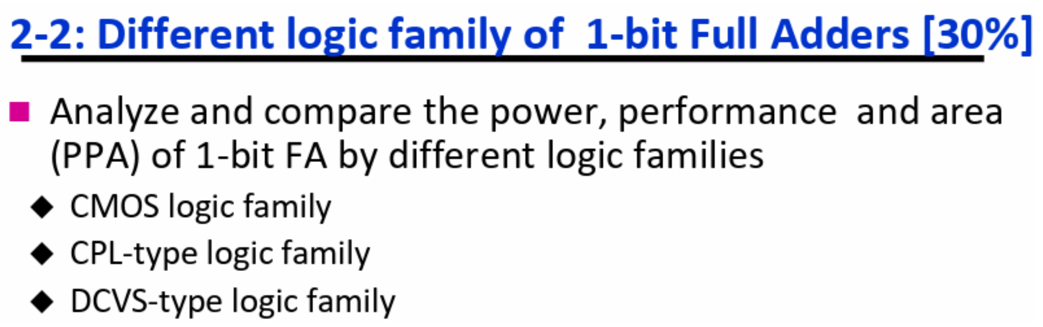
```
 tp15=   3.6505n  targ=   4.0637n   trig= 413.2229p
 tp16=   2.8292n  targ=   3.6389n   trig= 809.6615p
 tp17=  63.2692p  targ=   1.2763n   trig=   1.2131n
 tp18=  52.8521p  targ=   1.6630n   trig=   1.6102n
  **warning** (Ex2_1a.sp:103) .MEASURE tp19 never reached the target value
   tp19= failed       targ= not found    trig=    2.0107n
   tp20= failed       targ= not found    trig=    2.4085n
 tp21=  60.3879p  targ=   2.8710n   trig=   2.8106n
 tp22=  52.1913p  targ=   3.2615n   trig=   3.2093n
 tp23=  63.5149p  targ=   3.6737n   trig=   3.6102n
 tp24=  53.8015p  targ=   4.0626n   trig=   4.0088n
 tp25=  58.8897p  targ=   4.4689n   trig=   4.4100n
 tp26=  55.8243p  targ=   4.8646n   trig=   4.8087n
```


```
tp27=  60.9265p  targ= 472.9527p   trig= 412.0263p
tp28=  53.6635p  targ= 864.9080p   trig= 811.2445p
tp29=  59.9927p  targ=   2.0723n   trig=   2.0123n
tp30=  51.5243p  targ=   2.4629n   trig=   2.4114n
tp31=  59.9690p  targ=   2.8707n   trig=   2.8107n
tp32=  52.5339p  targ=   3.2618n   trig=   3.2093n
tp33=  57.4662p  targ=   3.6680n   trig=   3.6105n
tp34=  55.4582p  targ=   4.0644n   trig=   4.0090n
```


```
tp35=  48.9624p  targ= 459.0249p   trig= 410.0625p
tp36=  42.7420p  targ= 851.6007p   trig= 808.8587p
tp37=  55.7028p  targ=   2.8651n   trig=   2.8094n
tp38=  46.8113p  targ=   3.2559n   trig=   3.2091n
```


```
tp39=  61.0840p  targ= 473.0463p   trig= 411.9623p
tp40=  53.8134p  targ= 865.1266p   trig= 811.3131p
tp41=  60.1355p  targ=   1.2717n   trig=   1.2116n
tp42=  54.3841p  targ=   1.6654n   trig=   1.6111n
tp43=  57.2087p  targ=   2.0685n   trig=   2.0113n
tp44=  56.1456p  targ=   2.4669n   trig=   2.4107n
tp45=  56.8322p  targ=   4.0677n   trig=   4.0108n
tp46=  45.9040p  targ=   4.4568n   trig=   4.4109n
tp47=  49.9650p  targ=   5.2633n   trig=   5.2133n
tp48=  28.2301p  targ=   5.6386n   trig=   5.6104n
```

由上可知，最大 delay 在 tp5 時:即為 000 -> 011 上，A input 到 carry 的路徑為 worse case delay



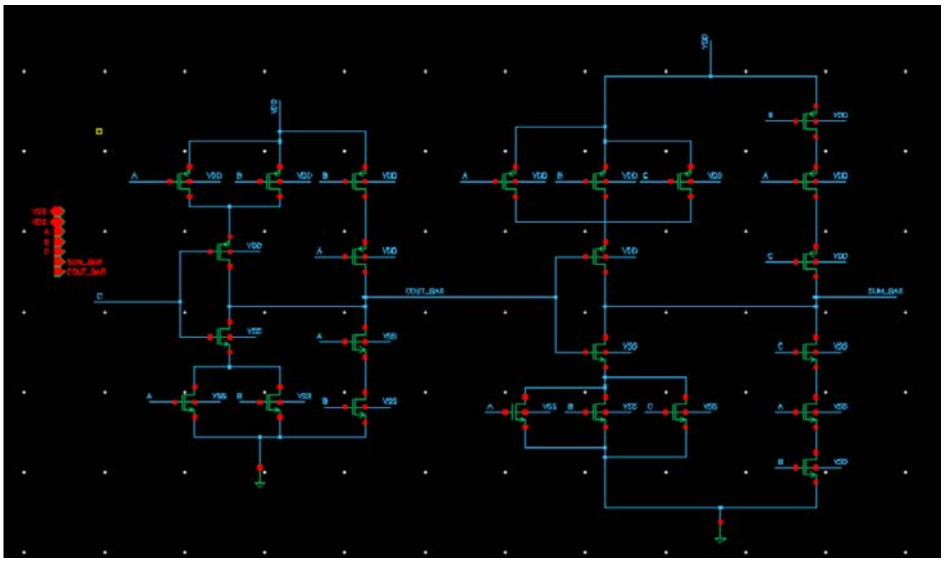
2-2: Different logic family of 1-bit Full Adders [30%]

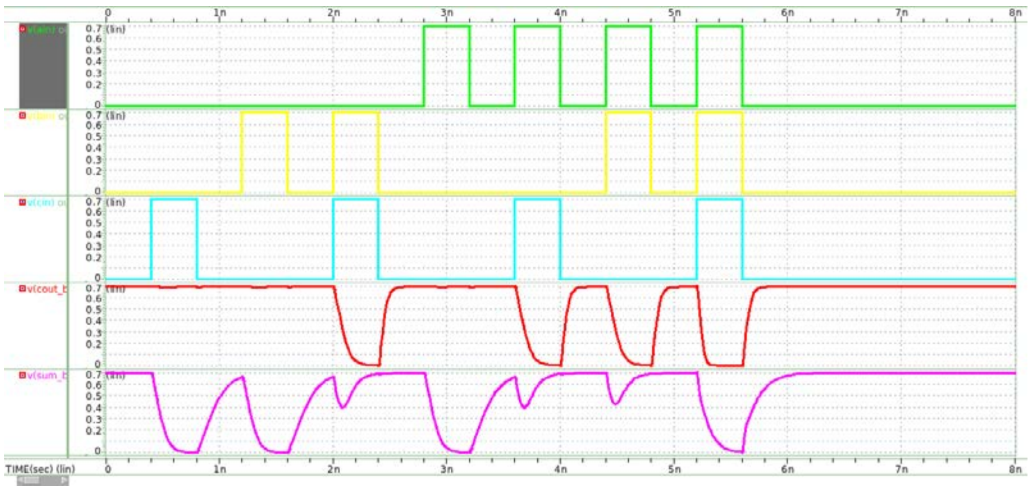
- Analyze and compare the power, performance and area (PPA) of 1-bit FA by different logic families
  - CMOS logic family
  - CPL-type logic family
  - DCVS-type logic family

算 power 時，我們看 average power；算 area 時，我們看各個 logic family 的電晶體個數；算 performance 時，我們看 power*delay，最小的代表 performance 最好，因為 delay 和速度有關，又速度和 performance 有關，delay 了話像上一題一樣找出每個加法器(每個皆 64 種可能)的 worst case Delay 作為基準點來判別。

以下為用 virtuoso 程式的會繪圖，並將它轉為.sp file:

CMOS:



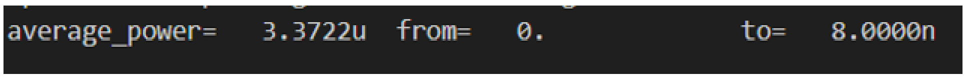


Area: 28T

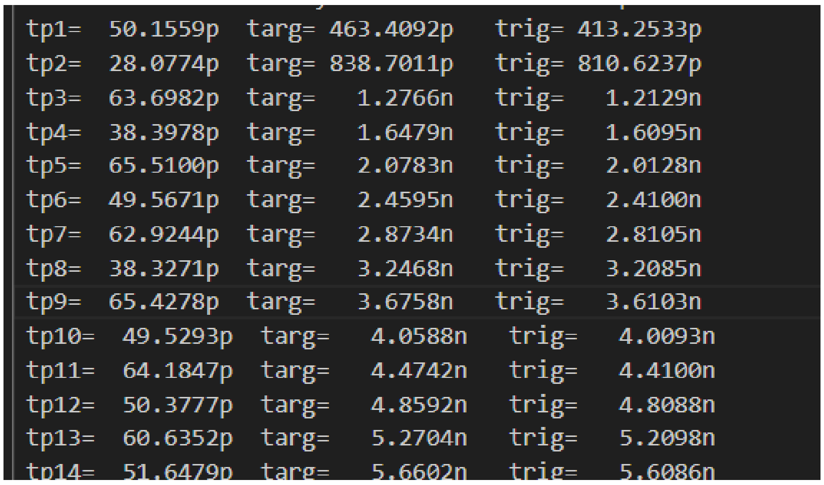
Delay:65.51ps (worst case 在 001 -> 010)

Performance: power*delay=2.209*10^(-16) J

Power: 3.3722uW

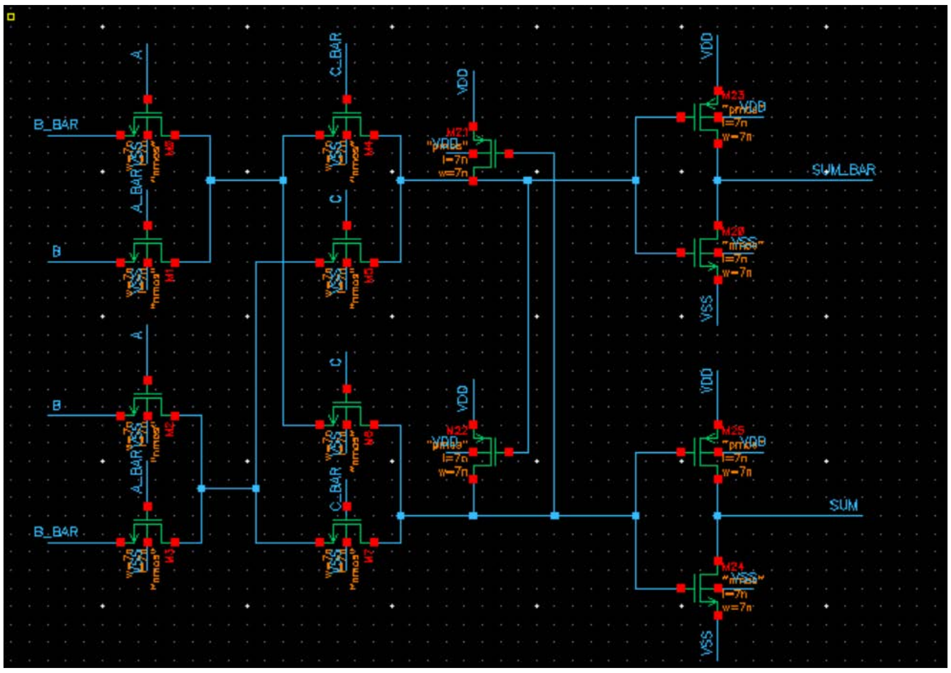


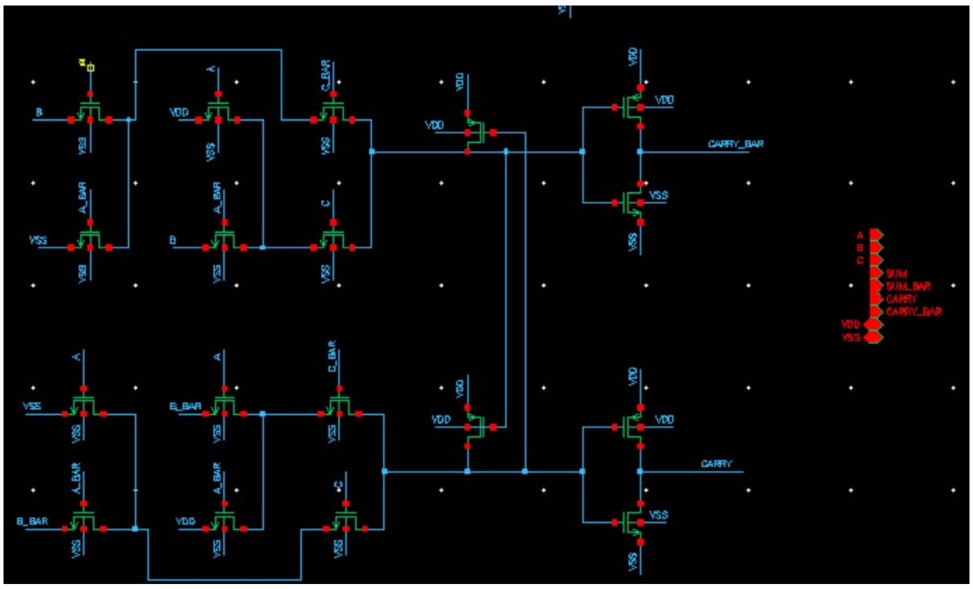
```
average_power=   3.3722u  from=   0.          to=   8.0000n
```

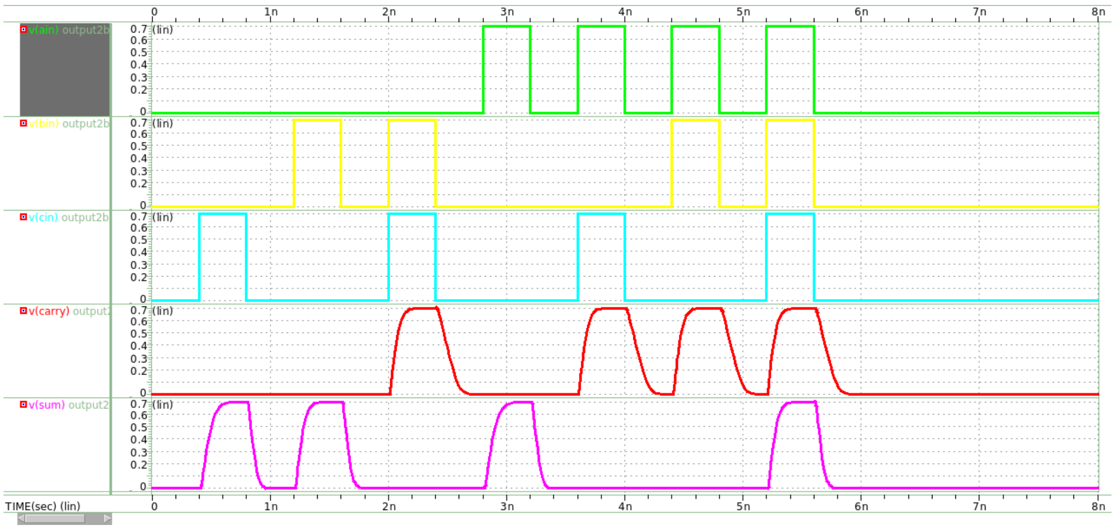


```
tp1=   50.1559p  targ= 463.4092p   trig= 413.2533p
tp2=   28.0774p  targ= 838.7011p   trig= 810.6237p
tp3=   63.6982p  targ=   1.2766n   trig=   1.2129n
tp4=   38.3978p  targ=   1.6479n   trig=   1.6095n
tp5=   65.5100p  targ=   2.0783n   trig=   2.0128n
tp6=   49.5671p  targ=   2.4595n   trig=   2.4100n
tp7=   62.9244p  targ=   2.8734n   trig=   2.8105n
tp8=   38.3271p  targ=   3.2468n   trig=   3.2085n
tp9=   65.4278p  targ=   3.6758n   trig=   3.6103n
tp10=   49.5293p  targ=   4.0588n   trig=   4.0093n
tp11=   64.1847p  targ=   4.4742n   trig=   4.4100n
tp12=   50.3777p  targ=   4.8592n   trig=   4.8088n
tp13=   60.6352p  targ=   5.2704n   trig=   5.2098n
tp14=   51.6479p  targ=   5.6602n   trig=   5.6086n
```

CPL-type:





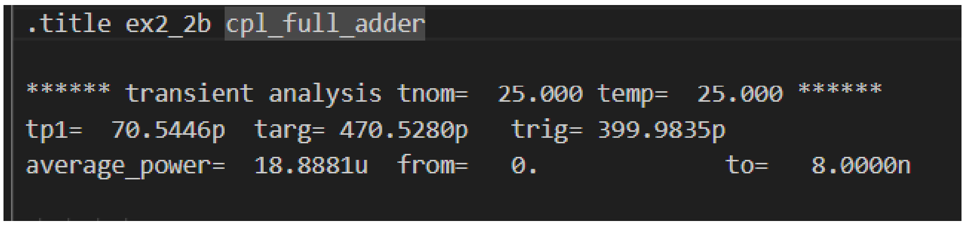


Area: 38T

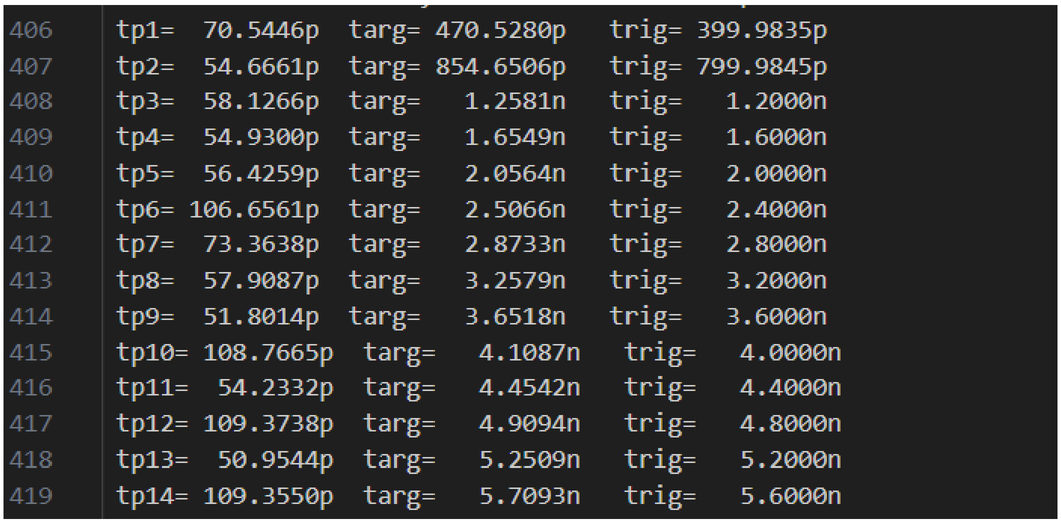
Delay: 109.3738ps (worst case 在 111 -> 000)

Performance: power*delay=2.066*10^(-15) J

Power: 18.8881uW

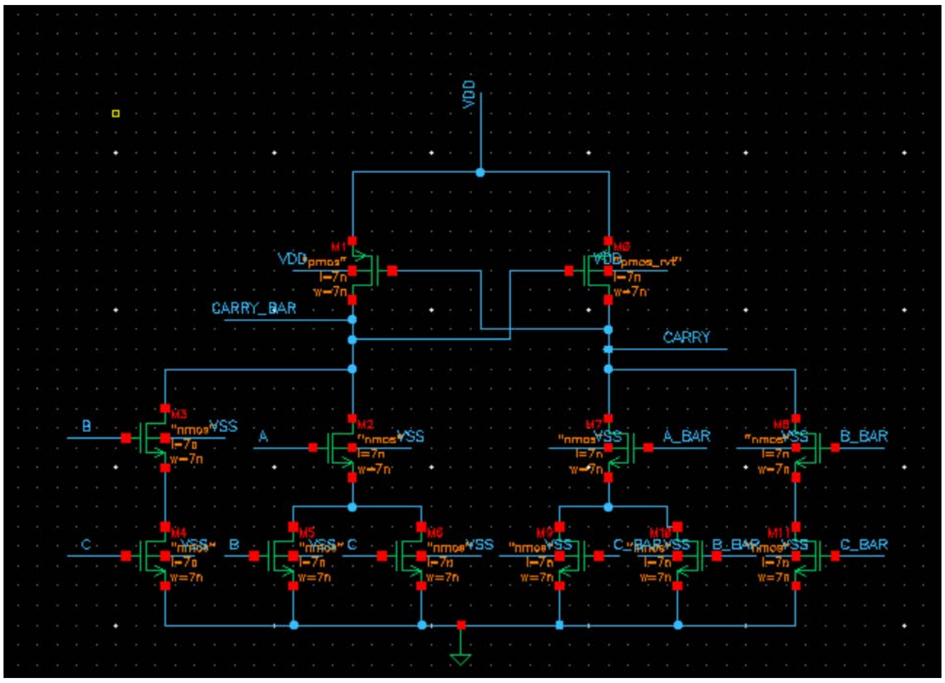


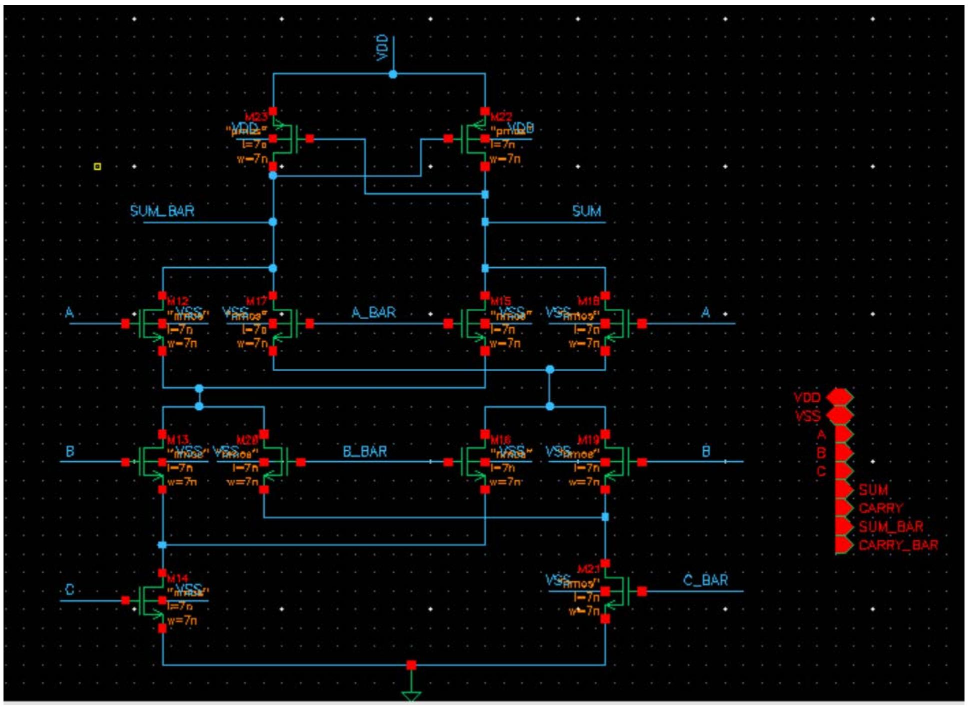
```
.title ex2_2b cpl_full_adder

****** transient analysis tnom=  25.000 temp=  25.000 ******
tp1=  70.5446p  targ= 470.5280p   trig= 399.9835p
average_power=  18.8881u  from=   0.            to=   8.0000n
```

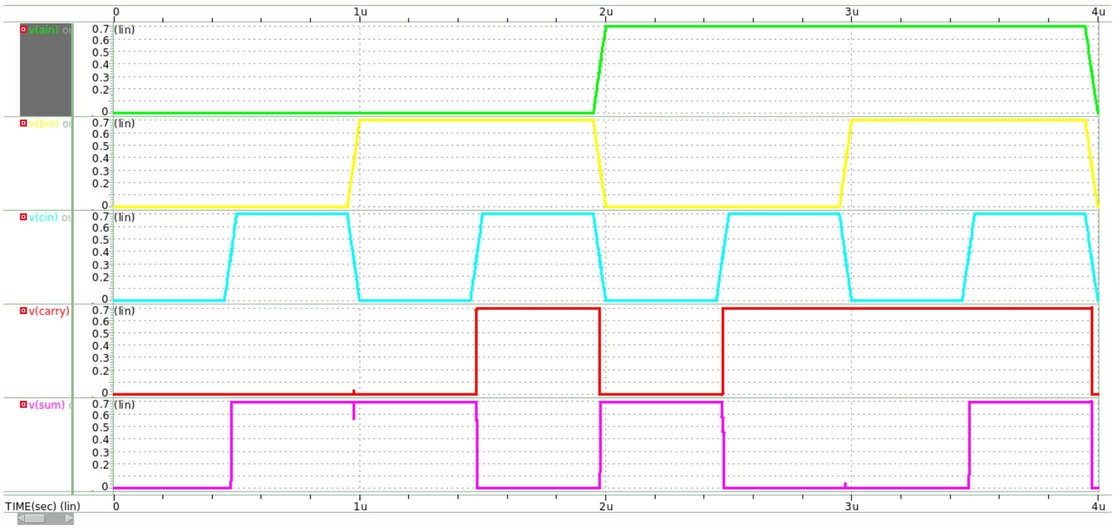


```
tp1=   70.5446p  targ= 470.5280p   trig= 399.9835p
tp2=   54.6661p  targ= 854.6506p   trig= 799.9845p
tp3=   58.1266p  targ=   1.2581n   trig=   1.2000n
tp4=   54.9300p  targ=   1.6549n   trig=   1.6000n
tp5=   56.4259p  targ=   2.0564n   trig=   2.0000n
tp6=  106.6561p  targ=   2.5066n   trig=   2.4000n
tp7=   73.3638p  targ=   2.8733n   trig=   2.8000n
tp8=   57.9087p  targ=   3.2579n   trig=   3.2000n
tp9=   51.8014p  targ=   3.6518n   trig=   3.6000n
tp10= 108.7665p  targ=   4.1087n   trig=   4.0000n
tp11=  54.2332p  targ=   4.4542n   trig=   4.4000n
tp12= 109.3738p  targ=   4.9094n   trig=   4.8000n
tp13=  50.9544p  targ=   5.2509n   trig=   5.2000n
tp14= 109.3550p  targ=   5.7093n   trig=   5.6000n
```

DCVS:





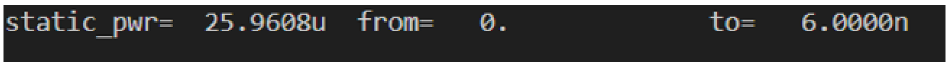


Area: 30T

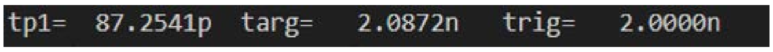
Delay: 121.1887ps   (worst case 在 010 -> 000)

Performance: power*delay=3.146*10^(-14) J

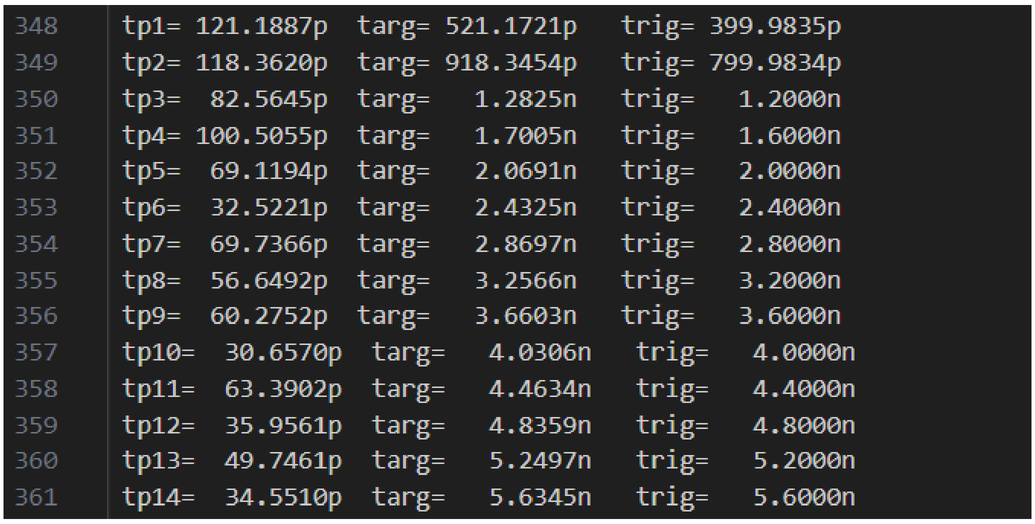
Power: 25.9608uW



```
static_pwr=  25.9608u  from=   0.           to=   6.0000n
```



```
tp1=  87.2541p  targ=   2.0872n   trig=   2.0000n
```


```
tp1= 121.1887p  targ= 521.1721p   trig= 399.9835p
tp2= 118.3620p  targ= 918.3454p   trig= 799.9834p
tp3=  82.5645p  targ=   1.2825n   trig=   1.2000n
tp4= 100.5055p  targ=   1.7005n   trig=   1.6000n
tp5=  69.1194p  targ=   2.0691n   trig=   2.0000n
tp6=  32.5221p  targ=   2.4325n   trig=   2.4000n
tp7=  69.7366p  targ=   2.8697n   trig=   2.8000n
tp8=  56.6492p  targ=   3.2566n   trig=   3.2000n
tp9=  60.2752p  targ=   3.6603n   trig=   3.6000n
tp10=  30.6570p  targ=   4.0306n   trig=   4.0000n
tp11=  63.3902p  targ=   4.4634n   trig=   4.4000n
tp12=  35.9561p  targ=   4.8359n   trig=   4.8000n
tp13=  49.7461p  targ=   5.2497n   trig=   5.2000n
tp14=  34.5510p  targ=   5.6345n   trig=   5.6000n
```

三個 logic family (1 bit) full adder 比較:

	CMOS logic family	CPL-type logic family	DCVS-type logic family
Power	3.3722uW	18.8881uW	25.9608uW
Delay	65.51ps	109.3738ps	121.1887ps
area	28T	38T	30T
Performance	2.209*10^(-16) J	2.066*10^(-15) J	3.146*10^(-14) J

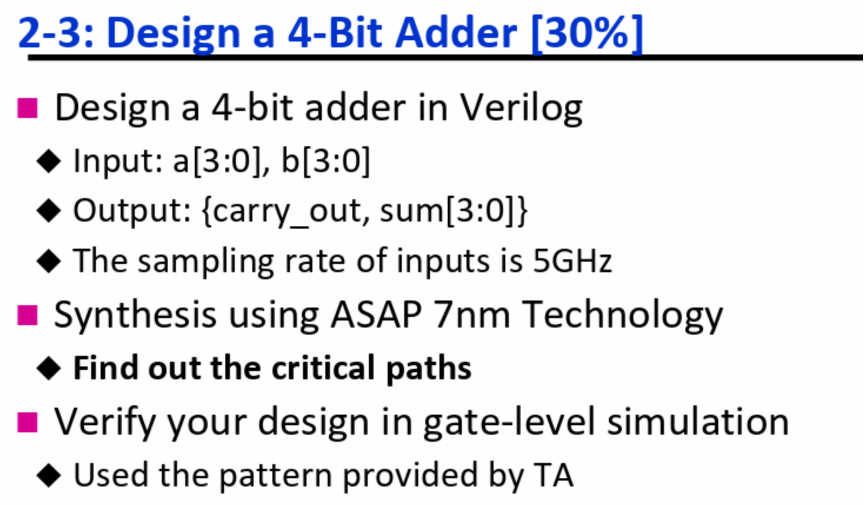
算 performance 時，我們看 delay 的時間和 power 的表現，因為 delay 和速度有關，又速度和 performance 有關，power 越小越好，所以 delay 小、power 越小，performance 就越好

結論:
Power: DCVS-type > CPL-type > CMOS
Performance: CMOS 優於 CPL-type 優於 DCVS-type
Area: CPL-type > DCVS-type > CMOS

以下為三種 logic family 的比較:
CMOS Logic (CMOS 邏輯)：
最壞延遲：CMOS 邏輯通常具有較短的延遲，因為它使用靜態 CMOS 閘來實現邏輯功能。它的延遲主要由輸入到輸出之間的兩個串聯 CMOS 反相器決定。
整體來看，CMOS logic family 的 1bit full adder 之 power 最小， performance 最好，area 最小，是三者中最佳的。而 CPL-type 的優點是功耗低 於 DCVS-type，且 performance 優於 CPL-type，不過 area 是三者中最大的。
電晶體數：28T

CPL-Type Logic (CPL 類型邏輯)：
最壞延遲：CPL 邏輯具有中間偏小的延遲，因為它使用傳輸閘來傳遞信號，減少了閘數量。
功耗：CPL 邏輯在操作時通常具有較高的功耗，因為傳輸閘在切換時消耗能量，但在閒置時功耗相對較低。
電晶體數：38T

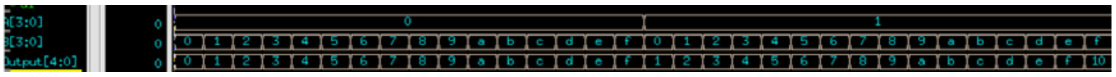
DCVS-Type Logic (DCVS 類型邏輯)：
最壞延遲：DCVS 邏輯通常具有較短的延遲，因為它採用預充電技術，並充分利用電壓擺幅（Voltage Swinging）來實現高速操作，但此例因為操作不是在高頻率下，所以相對其他兩個 logic 有較長延遲。
功耗：DCVS 邏輯在切換時可能有較高的功耗，但在閒置時功耗相對較低。它通常具有良好的功耗-性能平衡。
電晶體數：30T



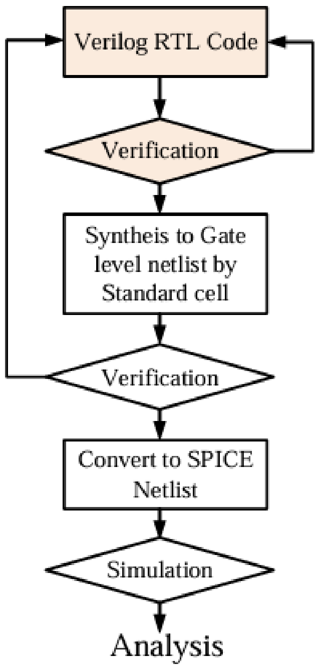
2-3: Design a 4-Bit Adder [30%]

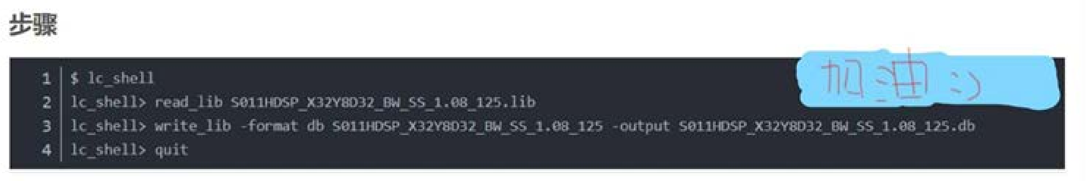
- Design a 4-bit adder in Verilog
  - Input: a[3:0], b[3:0]
  - Output: {carry_out, sum[3:0]}
  - The sampling rate of inputs is 5GHz
- Synthesis using ASAP 7nm Technology
  - **Find out the critical paths**
- Verify your design in gate-level simulation
  - Used the pattern provided by TA

首先，我按照助教給的指示，依循流程圖進行作業如下圖，根據助教的指示，會有 RTL_code，按照指示輸入指令後，可以驗證該 code 是否正確。
前模擬利用 nWave 去二次驗證 4-bit full adder 為正確的:


A[3:0] 0 | 0 | 1
B[3:0] 0 | 0 1 2 3 4 5 6 7 8 9 a b c d e f 0 1 2 3 4 5 6 7 8 9 a b c d e f
Output[4:0] 0 | 0 1 2 3 4 5 6 7 8 9 a b c d e f 1 2 3 4 5 6 7 8 9 a b c d e f 10

成功後，會輸出一熊貓圖。為了將 code 合成為 Gate level netlist，得將課堂提供的 lib 檔轉成 db 檔，步驟如下下圖，完成轉檔後，便可輸入指令，開始進行合成。

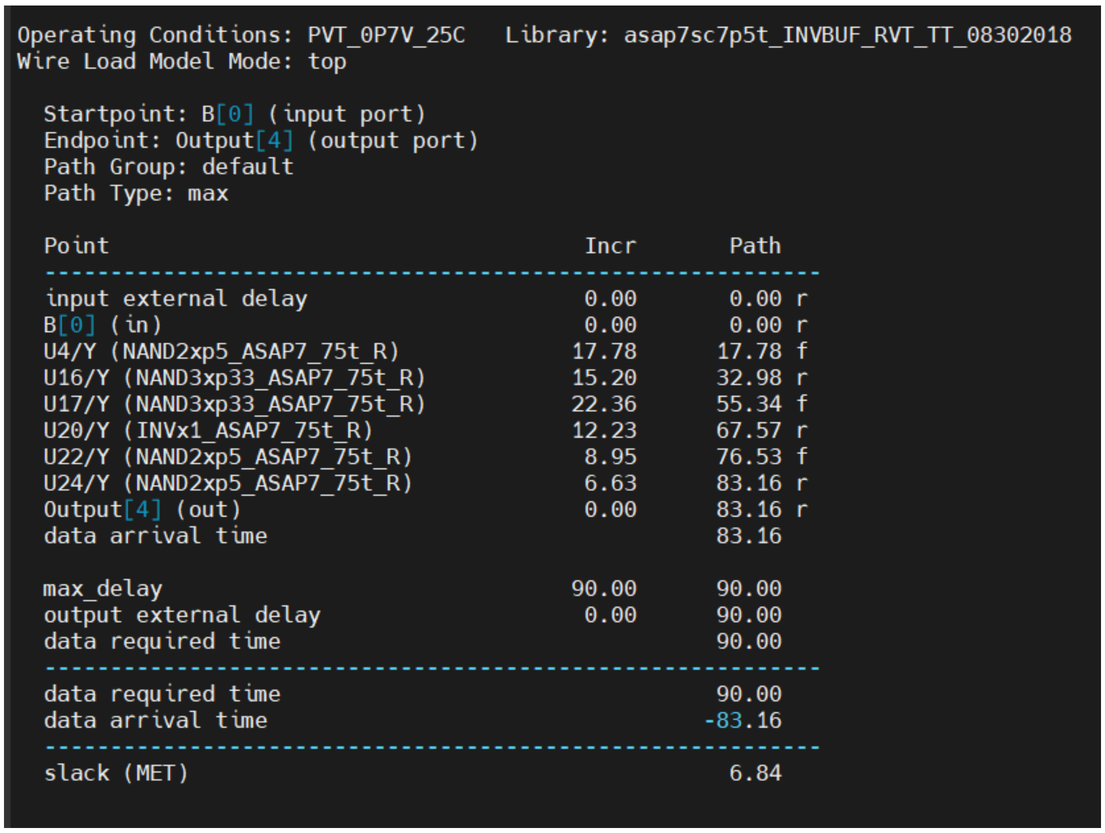




步驟

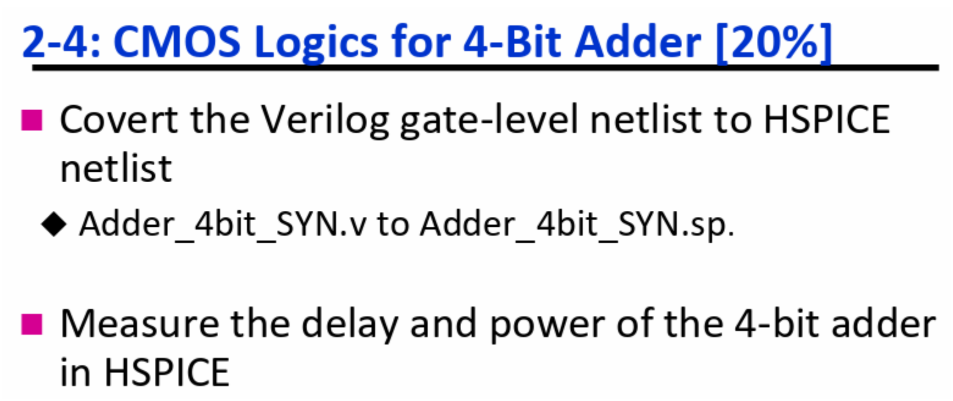
```
$ lc_shell
lc_shell> read_lib S011HDSP_X32Y8D32_BW_SS_1.08_125.lib
lc_shell> write_lib -format db S011HDSP_X32Y8D32_BW_SS_1.08_125 -output S011HDSP_X32Y8D32_BW_SS_1.08_125.db
lc_shell> quit
```

完成合成後，可以先在 syn.log 中看到合成後的資訊如下圖，便可以得知 critical path 為 B[0]到 Output[4](即為 carryout 的位置)，再次輸入驗證指令得到熊貓圖後，也就完成了 2-3 的步驟。


```
Operating Conditions: PVT_0P7V_25C   Library: asap7sc7p5t_INVBUF_RVT_TT_08302018
Wire Load Model Mode: top

  Startpoint: B[0] (input port)
  Endpoint: Output[4] (output port)
  Path Group: default
  Path Type: max

  Point                                        Incr       Path
  -----------------------------------------------------------
  input external delay                         0.00       0.00 r
  B[0] (in)                                    0.00       0.00 r
  U4/Y (NAND2xp5_ASAP7_75t_R)                 17.78      17.78 f
  U16/Y (NAND3xp33_ASAP7_75t_R)               15.20      32.98 r
  U17/Y (NAND3xp33_ASAP7_75t_R)               22.36      55.34 f
  U20/Y (INVx1_ASAP7_75t_R)                   12.23      67.57 r
  U22/Y (NAND2xp5_ASAP7_75t_R)                 8.95      76.53 f
  U24/Y (NAND2xp5_ASAP7_75t_R)                 6.63      83.16 r
  Output[4] (out)                              0.00      83.16 r
  data arrival time                                      83.16

  max_delay                                   90.00      90.00
  output external delay                        0.00      90.00
  data required time                                     90.00
  -----------------------------------------------------------
  data required time                                     90.00
  data arrival time                                     -83.16
  -----------------------------------------------------------
  slack (MET)                                             6.84
```

The critical path is from B[0] to Output[4]

The critical path delay time is 83.16



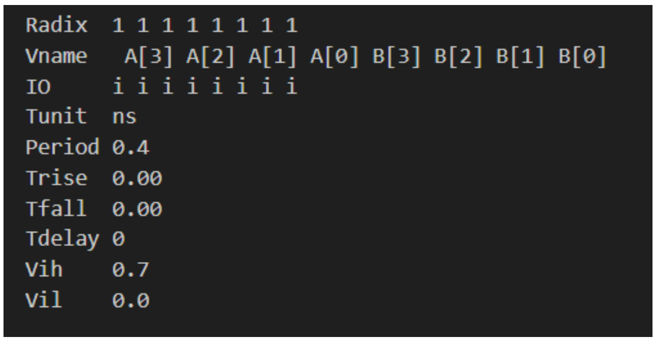
2-4: CMOS Logics for 4-Bit Adder [20%]

- Covert the Verilog gate-level netlist to HSPICE netlist
  - Adder_4bit_SYN.v to Adder_4bit_SYN.sp.
- Measure the delay and power of the 4-bit adder in HSPICE

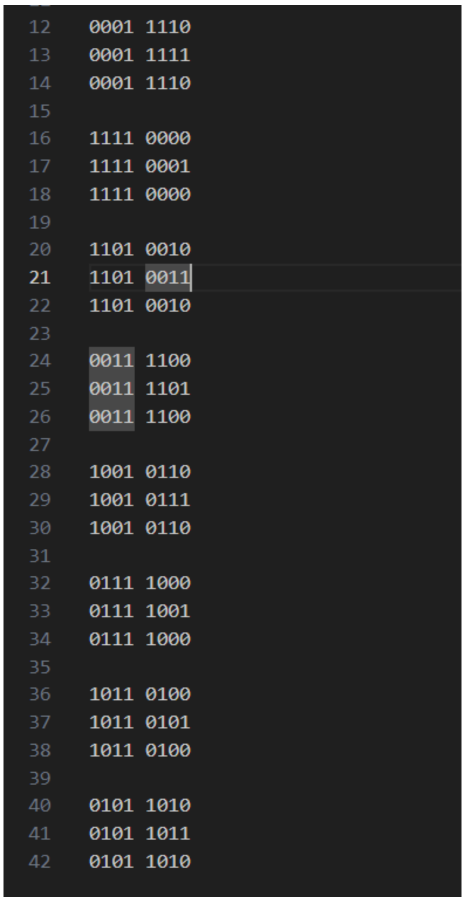
在 netlist 的資料夾中可以找到 Adder_4bit_SYN.v，其為合成後的結果。接著使用助教提供的指令將.v 檔轉成.sp 檔，再藉由提供的.py 檔補上接腳。

再將該子電路引用到最終要執行的.sp 檔，便可以開始設計 pattern。我設計的 Pattern_adder_4bit.vec 如下圖，設計以下八種 pattern 的原因是因為這八種狀況都會使所有 bit 進位，又由上題根據 critical path 知道 2-4 的 critical path 在 B[0] 到 output delay[4] 之間，所以我們只看 B[0]的變化對 output[4]的 delay，以符合在 critical path 的要求。

Pattern 設計如下:



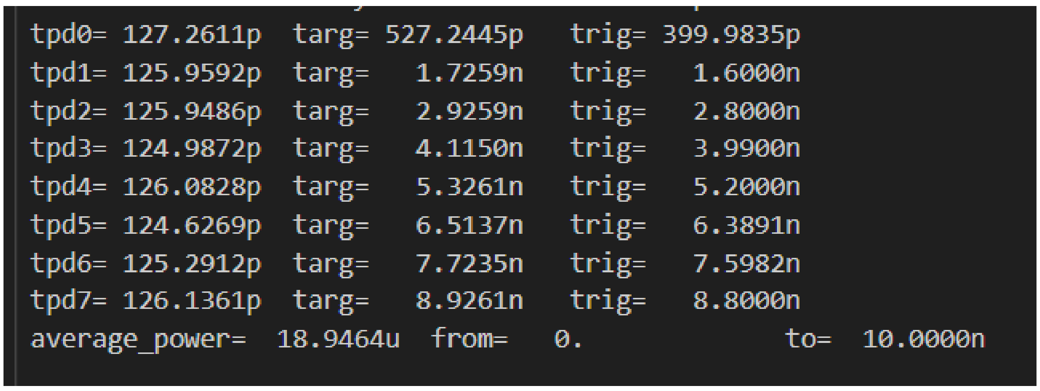
```
Radix  1 1 1 1 1 1 1 1
Vname   A[3] A[2] A[1] A[0] B[3] B[2] B[1] B[0]
IO     i i i i i i i i
Tunit  ns
Period 0.4
Trise  0.00
Tfall  0.00
Tdelay 0
Vih    0.7
Vil    0.0
```



```
0001 1110
0001 1111
0001 1110

1111 0000
1111 0001
1111 0000

1101 0010
1101 0011
1101 0010

0011 1100
0011 1101
0011 1100

1001 0110
1001 0111
1001 0110

0111 1000
0111 1001
0111 1000

1011 0100
1011 0101
1011 0100

0101 1010
0101 1011
0101 1010
```

使用包含 critical path 的 pattern

最後使用 Hspice 執行，便可得到以上八種狀況的延遲時間如下圖，而 worst case delay 的狀況便是 tpd0 即是從預設 A 的輸入為 0001，B 從 1110 變成 1111 的狀況，並得到 average power consumption 是 18.9464u。最後，我以下表來整理最後的結果。



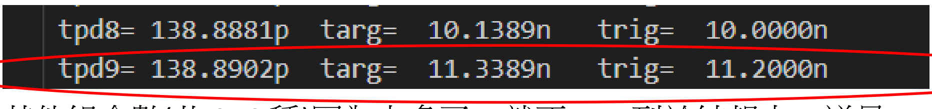
```
tpd0= 127.2611p  targ= 527.2445p   trig= 399.9835p
tpd1= 125.9592p  targ=   1.7259n   trig=   1.6000n
tpd2= 125.9486p  targ=   2.9259n   trig=   2.8000n
tpd3= 124.9872p  targ=   4.1150n   trig=   3.9900n
tpd4= 126.0828p  targ=   5.3261n   trig=   5.2000n
tpd5= 124.6269p  targ=   6.5137n   trig=   6.3891n
tpd6= 125.2912p  targ=   7.7235n   trig=   7.5982n
tpd7= 126.1361p  targ=   8.9261n   trig=   8.8000n
average_power=  18.9464u  from=   0.           to=  10.0000n
```

在 critical paths 的情況下，所有 critical paths 的延遲時間

A	B	tpd
0001	1110 -> 1111	127.2611p
1111	0000 -> 0001	125.9592p
1101	0010 -> 0011	125.9486p
0011	1100 -> 1101	124.9872p
1001	0110 -> 0111	126.0828p
0111	1000 -> 1001	124.6269p
1011	0100 -> 0101	125.2912p
0101	1010 -> 1011	126.1361p

各波形延遲時間比較表(紅色框即為固定 A 下之 worst-case delay)

以上為固定 A 下，改變 B[0]到 output[4]之 delay。接著，測 A 也跟著變動的 delay，最後發現 A、B[0]接跟著變動的 worst case delay 為當 A 由 0000->0011，且 B 由 1100->1111 時有 worst case delay: 138.8902p



```
tpd8= 138.8881p  targ=  10.1389n   trig=  10.0000n
tpd9= 138.8902p  targ=  11.3389n   trig=  11.2000n
```

其他組合數(共 256 種)因為太多了，就不一一列於結報中，詳見 Pattern_256_adder_4bit.vec