

面試大寶典

- 104 履歷每天都要上去點一下儲存，履歷才會是最新的，公司才看的到你
- 職缺需求是兩年的經歷也可以投看看
- 切記：公司再選你，你也在選公司。通常主管最後都會問你有沒有問題，這時候要提問薪資、福利跟工時的部分，也可以問一下今天面試的狀況，這樣子可以及時改進，然後面完不要覺得自己很爛，畢竟他們就是喜歡考倒你，主要是看你反應如何(假設遇到部會的，可以說這部分我不太清楚，可是我覺得...；主管會覺得你有會聽別人說，且會思考)
- 準備資料
 - PPT：可以用口試的改，內容放的所有東西都要很熟(參考學長姐的資料)(有些比較爛的就不要應放了，或者自己覺得有矛盾的也不要放，想個說詞過去，不要被抓到破綻)
 - 履歷：把自己的優勢放上去，例如：自傳(團隊合作、樂於解題)(參考仁甫跟彥丞)、英文成績(太低就不要放了)、作品集...
- PAPER 請自己閱讀一次，保留大概一個月的時間細細品嚐，DC 跟 ICC 有空要鑽研，要把所有的都搞懂，不要只有了解皮毛，要透徹
- 考題(請參考仁甫的筆記)，主管面試的時候問題有時候不會問得很清楚，你可以反問他或是直接問他是不是要考哪個東西，不會的話也可以說說你的解題思路(怎麼想的)，不確定也沒差，他只是想考你會怎麼想
- 面試一百題(上) <https://aijishu.com/a/10600000000124960>
- 面試一百題(下) <https://aijishu.com/a/10600000000124961>
- 產業價值鏈資訊平台(公司) <https://ic.tpex.org.tw/introduce.php?ic=D000>
- 仁甫筆記 <https://hackmd.home.morris13579.top:8443/s/Y6JOnNavI>
-

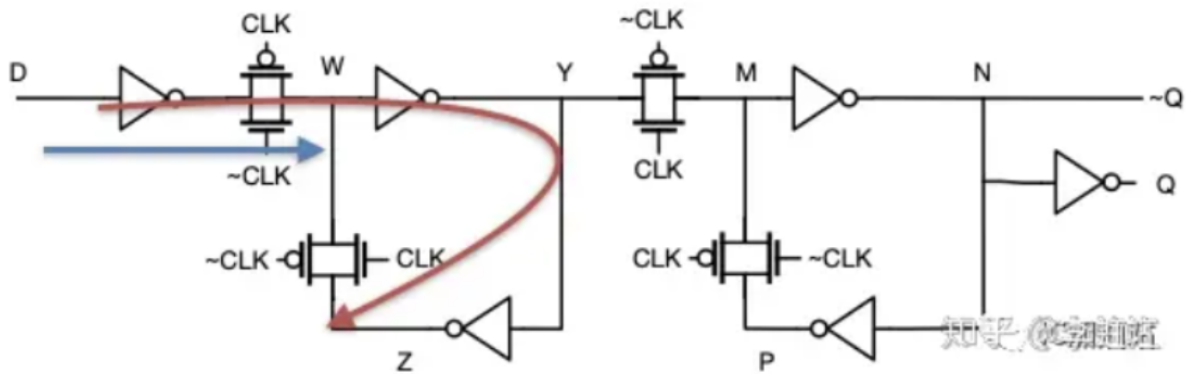
Q：為什麼 Static Timing Analysis 前不會跑出 hold time？

A：因為沒有 clock tree，且因為是理想。clock tree 的解釋：晶片裡面的 clock 是從外部輸入，每個 DFF 的 clock 路徑長度不同，會導致 RC 不同，RC 不同造成延遲不同。

Hold time 的定義是 $T_{hold} \geq T_{propagation\ delay}(min) + T_{clk_to_q}(min)$ ，猜測應該是編譯的時候將 Tpd 跟 Tclk_to_q 歸零；而 Setup time 的原因應該是在於 coding 沒寫好造成。猜測應該編譯是沒有將 Tset 歸零防止後面出錯。Setup time 的定義是 $\Delta T \geq T_{propagation\ delay}(min) + T_{clk_to_q}(min) + T_{set\ up}$ 。(ΔT 是工作週期)

Q：setup time、hold time？

A：setup time 定義 $\Delta T \geq T_{propagation\ delay}(min) + T_{clk_to_q}(min) + T_{set\ up}$ ，資料不能 sample(上升或下降)的時候有改變，(有改變就會造成值會不同，不同就會不穩定，造成 metastable)。hold time 定義 $T_{hold} \geq T_{propagation\ delay}(min) + T_{clk_to_q}(min)$ ，資料維持的時間不夠，會造成吃到的值有可能是錯的。



其中的Latch能夠有儲存資料的能力靠的是上面的背對背的反相器，而這個背對背的反相器需要時間來儲存資料。因此可以得出以下

setup time: 在clock的posedge到來前D->Z所需的時間，若Z的值還沒穩定時D就變化，那這個背對背的反相器就無法存住值

hold time: 第一個transmission gate關閉需要的时间，關閉的期間D->W要保持穩定，這樣在transmission gate關閉後W穩定才不會導致背對背反相器鎖住的值發生變化

因此若不滿足setup/hold time要求時，Q的值可能不是原本我們輸入的D

Q：甚麼時候會發生 setup time、hold time？

A：setup time 當 A 比較長的時候

Q：甚麼時候會發生 setup time、hold time？

A：setup time 加脈衝（）。hold time 加 DFF 或 BUF

Q：甚麼是 metastable(5W1H)？

A：setup time、hold time 不滿足會發生 sample 到的值不是正確的，他的值會是 0 跟 1 之間的未定值(有可能是 0、1、0.7VDD 或 0.3DD 中間的未定值)。

Q：有 metastable 怎麼辦？很多 bit 怎麼辦？

A：metastable 沒辦法完全消除，等一段時間(一兩個週期)會自己回到穩態，減少的话就是多敲一級，也就是加 DFF，從 MTBF 中可以知道會隨著 DFF 增加機率會變很小。多 bit 的話要用 async FIFO 或者 Multi-Cycle Path (MCP) formulation (要去看 paper，其他主管一定會再繼續往下問)

async FIFO 定義是先進先出的記憶體，非同步代表需要用指標指向現在 write 或是 read 的地方，可能兩邊的速度不同，所以需要使用到非同步。指標建議使用 gray code 因為一次只會變化一個 bit，降低 metastable 的可能性。

非同步FIFO (Asynchronous FIFO)

FIFO概念

- 會有read_ptr和write_ptr，當資料寫入時write_ptr+1，讀取資料時read_ptr+1
- write_ptr領先read_ptr代表FIFO寫入已經滿了
- read_ptr領先write_ptr代表FIFO空了
- 問題
 - write_ptr與read_ptr在不同的時鐘域，需要將結果同步到另一個時鐘域，才能判斷FIFO空了還是滿了
 - write_ptr與read_ptr同步到不同另一時鐘域會遇到multi bits的CDC問題

格雷碼(Gray code)

- 特點
 - 每次只會變化一個bit
 - 當多條訊號線需要使用CDC傳送到另一個clock域時，像是計數器的值，如果沒有使用格雷碼，那麼就會需要使用Hand Shake的方式，而使用格雷碼就可以直接使用同步器
- 二進制轉格雷碼
 - $G_n = B_{n+1} \oplus B_n$
- 格雷碼轉二進制
 - $B_3 = G_3$
 - $B_2 = G_3 \oplus G_2 = B_3 \oplus G_2$
 - $B_1 = G_3 \oplus G_2 \oplus G_1 = B_2 \oplus G_1$
 - $B_0 = G_3 \oplus G_2 \oplus G_1 \oplus G_0 = B_1 \oplus G_0$

格雷碼計數器

- 風格1

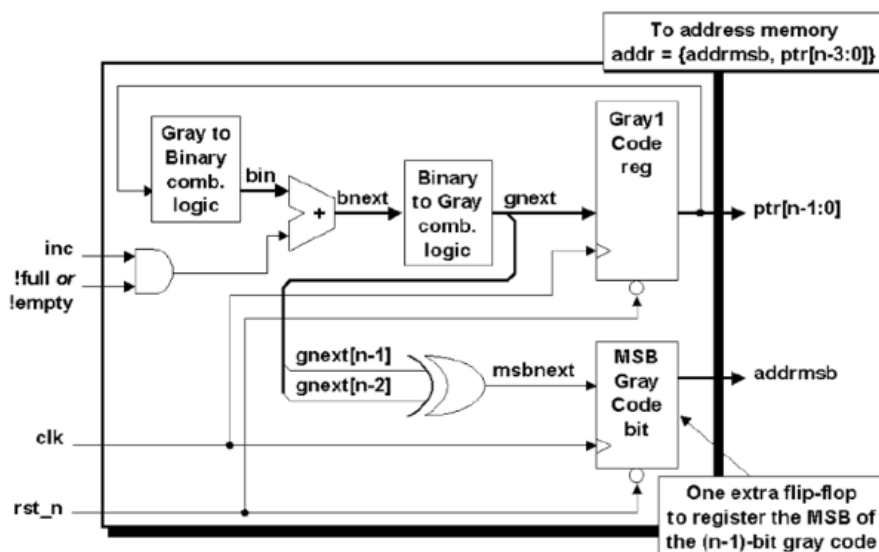


Figure 3 - Dual n-bit Gray code counter block diagram - style #1

- 每次加法前將目前的格雷碼轉回二進制，加完後再轉回格雷碼

- 風格2

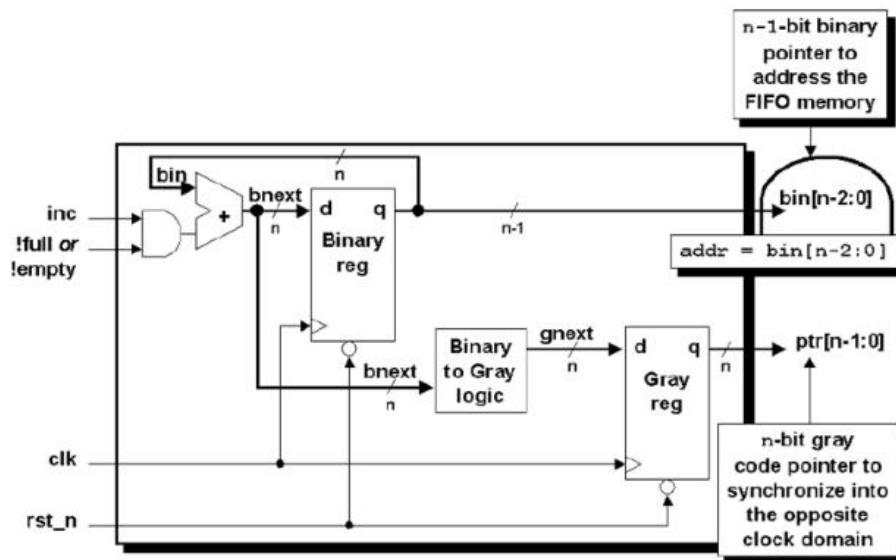


Figure 4 - Dual n-bit Gray code counter block diagram - style #2

Q：Inv 怎麼加速？

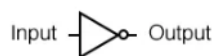
A：因為溫度跟電壓

Q：為什麼 DC 有 best case 跟 worst case？

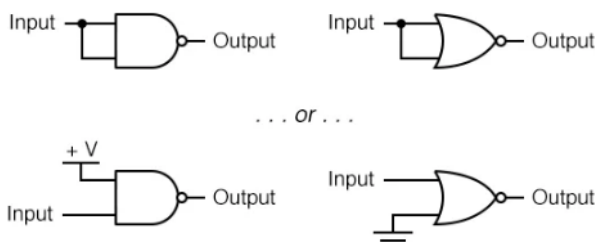
A：PVT 會影響到 best case 跟 worst case；VT 會影響到在 DC 下的 best case 跟 worst case。製程(P)，因為製程的技術的進步，可以讓導線變短，可以讓 RC(電阻、電容)下降；電壓(V)，因為電壓提高會造成電流變大，電流變大電容充充電速度變快，RC 延遲變小；溫度(T)，不同製程在溫度改變時，會有不同的特性。18 製程溫度越高 performance 越好

Q：給出一個電路問 function 並寫出 verilog、使用 NAND / NOR / MUX 組出反向器

NAND、NOR完成NOT功能



Input	Output
0	1
1	0



- 以上電路皆可以完成NOT的功能，主要的差異是如果將輸入接在一起時，影響前一級扇出數 (Fanout)

A：

同步器是為了消除 metastable(以下是同步器的 verilog)

```
always@(posedge clk)
```

```
    D1 <= in
```

```
    D2 <= D1
```

Q：CDC 單 bits 與多 bits 怎麼處理，寫出同步器的 verilog

A：hand shake 裡面會用到脈衝

Q：同步 reset 與非同步 reset

A：同步 reset 是跟 clock 一起，非同步 reset 是隨時都可以 reset(clr)

Q：CDC 多 bits 如何解決？

A：如果多 bits 同樣使用同步器，可能會發生資料在不同時間同步過去，造成資料錯誤。

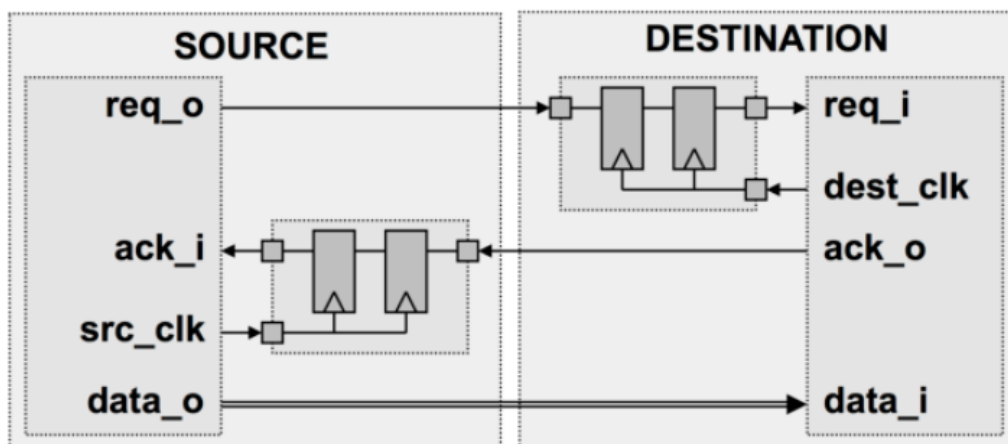
解決方式：Hand Shake、Async FIFO

CDC多bits如何解決

- 如果多bits同樣使用同步器，可能會發生資料在不同時間同步過去，造成資料錯誤
- 解決方式
 - Hand Shake
 - Async FIFO

Hand Shake

方法



- 當資料準備好的時候，利用 req_o 同步到對面的clk domain B，這時clk domain B就知道A已經將資料準備好了，並且這時候A的 data_o 會完全靜止不動，等到B已經取樣資料完畢，利用 ack_i 告訴clock domain A資料已經取樣完畢。

缺點

- 增加資料要傳遞得時間

非同步FIFO (Asynchronous FIFO)

FIFO概念

- 會有read_ptr和write_ptr，當資料寫入時write_ptr+1，讀取資料時read_ptr+1
- write_ptr領先read_ptr代表FIFO寫入已經滿了
- read_ptr領先write_ptr代表FIFO空了
- 問題
 - write_ptr與read_ptr在不同的時鐘域，需要將結果同步到另一個時鐘域，才能判斷FIFO空了還是滿了
 - write_ptr與read_ptr同步到不同另一時鐘域會遇到multi bits的CDC問題

Q：IC design flow？

A：1.規格 2.RTL 設計 3.Synthesis 合成 4.APR 佈局

Q：怎麼看溢位？

A：要先看是不是二的補數，是的話就是 $OF = C_n \oplus C_{n-1}$

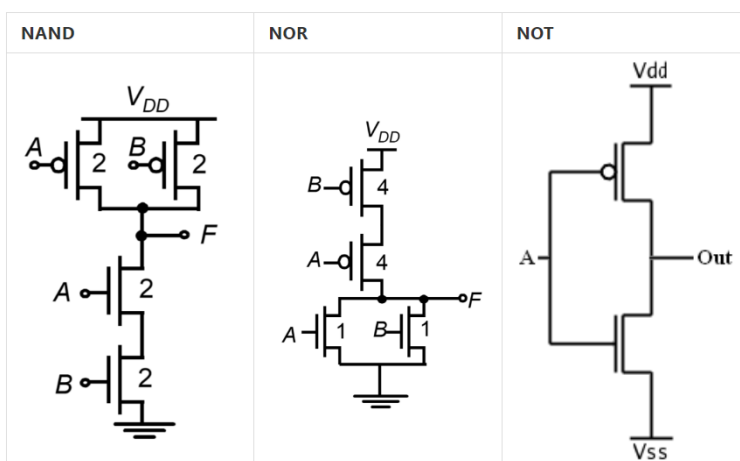
Q：如何避免 clock skew/latch？

A：clock skew：長 clock tree 的時候會自己長 delay buff，避免不了。latch：block 的時候 if/else 跟 case 要寫滿(養成好習慣，平常就寫滿)；寫 initial condition(歸零的意思)

Q：cell library 的 hold time 有什麼特色？

A：它是對信號的最短持續時間的限制，以確保數位電路的**正確性**和**可靠性**。它是在設計數位電路時需要考慮的重要因素，以確保信號在整個電路中能夠得到準確的傳遞。在使用 cell library 時，需要根據不同的應用場景和需求來選擇合適的 hold time，以維護數位電路的性能和穩定性。

Q：用 PMOS 和 NMOS 組 NAND & INV？



A：

Q : Xor truth table ?

A :

A	B	C
0	0	0
0	1	1
1	0	1
1	1	0

群聯

Q：LEC 是在做什麼？

A：邏輯等效檢查，比較 RTL code 跟優化完的 netlist 是不是等效。

Q：DC 怎麼變 netlist？

A：translate(轉譯)、mapping(每個元件擺放的對應位置)，DC 吃 RTL 檔，會出來 gate-level netlist 檔；DC 裡面的 netlist 重點在程式碼跟邏輯元件之間對應的關係，translate 會吃 Designware library(DW，在新思提供的 models 資料庫，例如：Adder、Multipliers、Dividers)找元件庫；mapping 會吃 technology library，看用什麼製程(製程會提供的 models，例如：AND、OR、FF)。

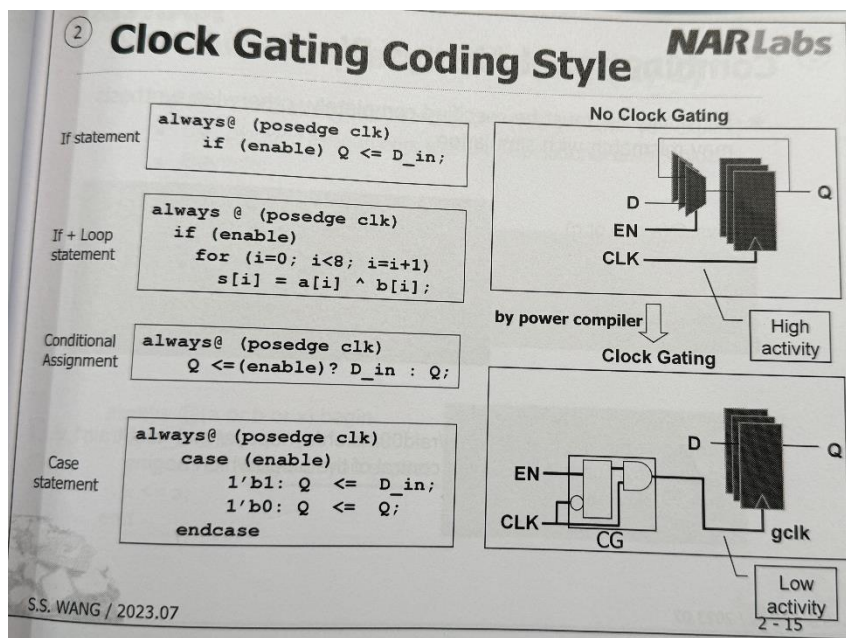
補充：ICC。Place(佈局)、CTS(長 clock tree)、Route(繞線)，ICC 吃 gate-level netlist 檔，會出來 gds 檔(佈局圖)，ICC 裡面的重點在將 netlist 轉成實際上的 layout，繞線跟設定 power。

Slack(餘裕)：要求時間

Q：coding(RTL)怎麼寫可加 clock gating？(DC 課本第 2-15)

A：加 clock gating 是為了減少 power，讓 FF 可以不用每個 edge 都要 sample 資料。

- 法一(打 code)：加一個致能訊號(enable)
- 法二：法一不夠好的話，在 DC 合成時用指令將 clock gating 變成使用 ICG(integral clock gating，積體/整合型 clock gating) cell 的方式合成。



Q：NC-Verilog、Calibre、Virtuoso、Verdi？

A：

- NC-Verilog 是跑模擬(DC 跟 ICC 都會做)，將 testbench 跟 netlist 去模擬電路的結果。
- Calibre 是會做 DRC、LVS、PEX。
 - DRC 是設計規範驗證（英語：design rule check，DRC）可修正並檢驗佈局（layout）是否符合設計規範，但 DRC 無法保證在佈局完全符合設計規範的情況下，線路依舊維持設計者的預期，而 LVS 則是這個階段的最適合的解決方案。
 - LVS 是電路佈局驗證（英語：Layout versus schematic，LVS）是一種電子設計自動化（英語：electronic design automation，EDA）工具，其功能為驗證特定積體電路與其原始電路設計之間的差異有無異常。
 - PEX 是萃取電阻、電感、電容(會算出他們的值)，就可以做 post sim。陳的猜測可能是 full custom 才會用到。
- Virtuoso 是畫 layout
- Verdi 是拿來看 nWave 的，但他實際有很多功能可以使用。
- NC-Verilog 會將 testbench 跟 netlist 去模擬電路的結果，再用 Verdi 的 nWave 去看波形。


瑞昱

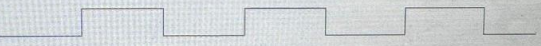
Q：除三電路寫 code，兩種思路 duty cycle 50，或是不管 duty cycle？

Clock divider


- Please coding a clock divider (Divide by 3).

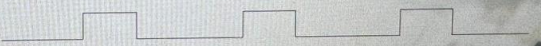
Case1

Ori 

Div3 

Case2

Ori 

Div3 

www.realtek.com.tw -3- Copyright © 2011 Realtek Semiconductor Corp.

A：duty cycle 50 看仁甫筆記；不管 duty cycle 用計數器寫

Q：FIFO(主管說如果不會可以換下一題)

FIFO

- Please design a FIFO logic, the depth is 4 entry.
- Code requirement:

```
module fifo_ctrl (  
    clk, rstn, push, pop,  
    din, dout, full, empty  
);  
  
    *****  
    *****  
  
endmodule
```

www.realtek.com.tw -4- Copyright © 2011 Realtek Semiconductor Corp.

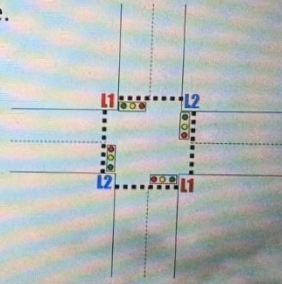
A：我回他，我可以試看看(心裡 OS：這是陷阱 這是陷阱 這是陷阱)

Q：紅綠燈題(主管說很多人不會寫這題，死在這)(後來只有要我先寫前半段，設參數那邊)(寫完後問我要不要在繼續寫)

Traffic light control

- Please design a traffic light control logic, refer right figure.
- Code requirement as below:

```
module traffic_ctrl (  
    clk, rstn, light_1, light_2  
);  
    .....  
    .....  
endmodule
```



A：我回他，面試時間還夠嗎(擔心一下他之後有沒有其他事)(心裡 OS：這是陷阱 這是陷阱 這是陷阱)如果還有時間我可以繼續寫(他說好)(心裡 OS：果然是陷阱)