

VLSI Final Project Report

Group 1: 110060027 朱豐蔚、110060020 戴佑丞

一、Design

1. 設計目標

在正式進入設計前，我們想要先定下來整體設計上的方針與目標，讓後續的各項設計中遇到 trade off 的情況時，我們能夠更好的且合理的做出選擇。

$$Performance = \text{Max operation frequency}^2 \div \text{area}^2 \div \text{power}$$

由最終的 FoM 公式，可以知道若想要追求較佳的 performance，可以從三個方向去下手，分別是增加頻率、減少面積與減少功耗，但這三點彼此並非獨立的，三者會互相形成拮抗、trade off 的關係，舉例來說，追求頻率會使得功耗、面積都有所上升，因此在一開始就定下主要目標會是必要的過程。

首先分析三項彼此之間的關係：

(1) 頻率與功耗：

由 CMOS 的 dynamic power 公式可以推知， $P \propto CV_{DD}f$ ，因此升高頻率的同時也會增加功耗，整體呈現正比的關係。

(2) 頻率與面積：

若要增加頻率，意味著 RC delay 要降低，也就是要降低 R 與 C，因此首要目標就是讓所有 L 都最小，調成 0.18 um，並且增加 W 去降低 R 的量值。因此可以知道，若想要增加頻率，就必須要從增加 W 下手，而增加 W 也就是增加面積，兩者是成正相關的，但 N 倍的面積很可能無法做到 N 倍的頻率。

結合上述兩點，在電路架構不變的情況下，若想提高頻率為兩倍，就會使得功耗提高為約兩倍、面積可能也會相對提高超過兩倍，這表現在 performance 上反而造成 FOM 變為小於 0.5 倍，因此一味提高頻率可能不是最佳 FOM 的設計方式。

換句話說，因為頻率約正比於功耗，因此對於 FOM 而言，FOM 正比於頻率，而 FOM 反比於面積的 2 次方，故我們選擇從面積入手，盡可能的在能夠滿足 SPEC 的功能下，設計出越小的面積，並強調畫 layout 時的空間利用性，我們在此 project 中會著重在這兩部份的處理上。

最後才是求頻率與功耗的表現，自此定完整個電路的設計目標。

2. 電路架構

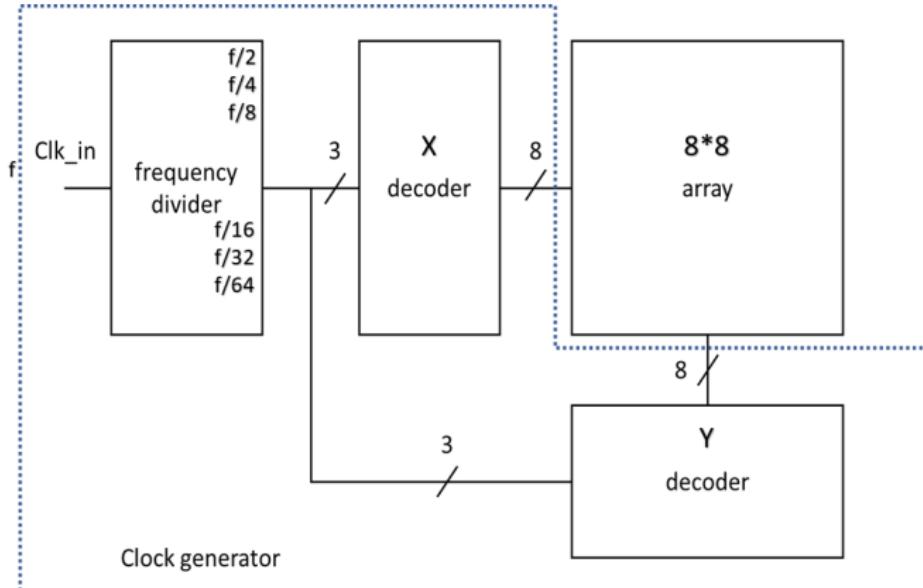


fig1. System block diagram

首先考慮整個系統的架構，如上圖，總共需要設計一個 frequency divider 和兩個 decoder，decoder 分別提供 bit line 與 word line 的 decoder，因此可以設計不相同的架構與尺寸，此點於後續再進行討論。在這一部分中，我們先著重在要使用哪些 gate 與元件來達成 frequency divider 和 decoder 的功能性。

首先先設計 decoder，其中 decoder 需要將 input 訊號 enable、A、B、C，轉換成輸出的 D0 ~ D7，先列出輸出與輸入的關係式如下：

$$\begin{aligned}
 D0 &= \text{enable} \& \sim A \& \sim B \& \sim C \\
 D1 &= \text{enable} \& \sim A \& \sim B \& C \\
 D2 &= \text{enable} \& \sim A \& B \& \sim C \\
 D3 &= \text{enable} \& \sim A \& B \& C \\
 D4 &= \text{enable} \& A \& \sim B \& \sim C \\
 D5 &= \text{enable} \& A \& \sim B \& C \\
 D6 &= \text{enable} \& A \& B \& \sim C \\
 D7 &= \text{enable} \& A \& B \& C
 \end{aligned}$$

可以看出在 decoder 中 $\sim A$ 、 $\sim B$ 、 $\sim C$ 也為輸入的一部分，若傳入 decoder 時只給入 A、B、C，還需要額外 3 個 inverter 去做反轉，因此我們設計希望讓 frequency divider 直接給入完整的 A、B、C、 $\sim A$ 、 $\sim B$ 、 $\sim C$ 訊號，來節省電晶體的用量，以利後續的 layout 能畫出更小的面積。

接著考慮 and 架構實作上可以有許多方式來達到以上的功能，為了追求面積能夠最小，因此下列可能的實行方式中，會以電晶體使用數量的角度來考慮與選擇：

(1) 使用 8 顆 and4 來實行，and4 由 nand4 加上 inverter 組成：

一個 decoder 總共會使用 $8 \times (8 + 2) = 80$ 顆電晶體。

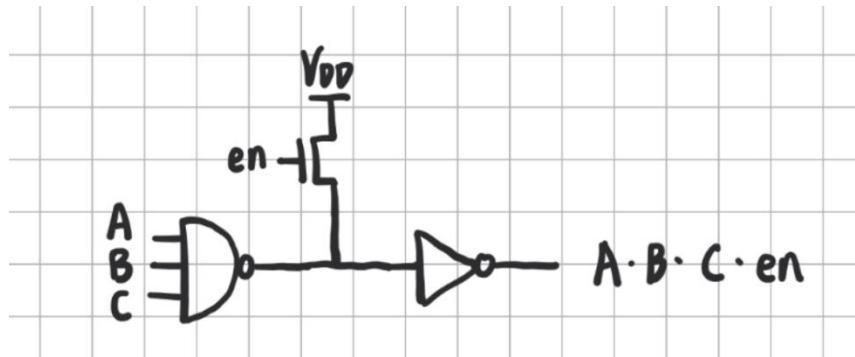
(2) 使用 8 顆 and4 來實行，and4 由兩顆 nand2 加上 nor2 組成：

一個 decoder 總共會使用 $8 \times (2 \times 4 + 4) = 96$ 顆電晶體。

(3) 使用 8 顆有 enable 功能的 and3 來實行

一個 decoder 總共會使用 $8 \times (6 + 1 + 2) = 72$ 顆電晶體。

因此在上述三種選擇中，我們選擇使用 8 顆有 enable 功能的 and3 來實行 decoder 的設計，下列將會說明如何實現有 enable 功能的 and3。



如上圖，可以看到 enable 紙入一顆連接中間的 pMOS，當 enable 等於 1 時，pMOS 會關閉，代表此時輸出為 A、B、C 三者的 and；而當 enable 為 0 時，pMOS 會開啟，此時會將中間連接的電位藉由 VDD 充電成 1，inverter 出來後的結果就會被強制拉成 0，實現 enable 的功能。

使用此架構除了面積外還有另外的優勢，第一個是速度，nand3 的 g 會比 nand4 的 g 小，logic effort 上有優勢，加上仔細計算由兩顆 nand2 加上 nor2 組成的 logic effort 表現也會較差，因此採用有 enable 功能的 and3 有速度上的優勢。

再者就是功耗，由於多加的 pMOS 只有在 enable 變更時有功耗，因此不會影響後續的正常 decoder 範圍功耗表現，功耗表現也會較佳。

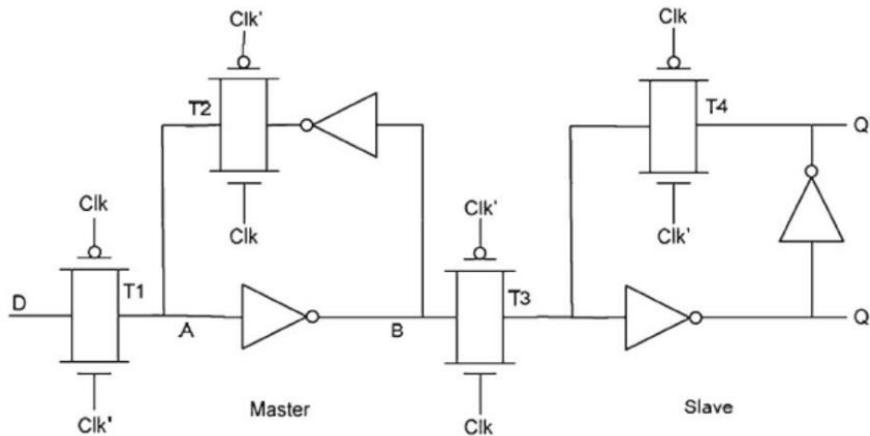
結合面積、功耗、頻率，有 enable 功能的 and3 都有優勢，因此我們選用此種架構作為 decoder 的設計。

接著討論 frequency divider 的設計架構，前面提到為了節省電晶體的用量，我們希望傳入 decoder 的訊號能夠有完整的 A、B、C、~A、~B、~C 訊號，若只給入 A、B、C，還需要額外 3 個 inverter 去做反轉，因此 frequency divider 的目標為將傳入的 clk_in 訊號除頻出 $f_2 \sim f_{64}$ 的正反訊號，並且能有 reset 的功能。

由於 frequency divider 是一個 sequential circuit，因此我們需要具有記憶性的電路，如 flip-flop 或是 latch，而為了達到除頻的功能，應選擇 flip-flop 來實作。

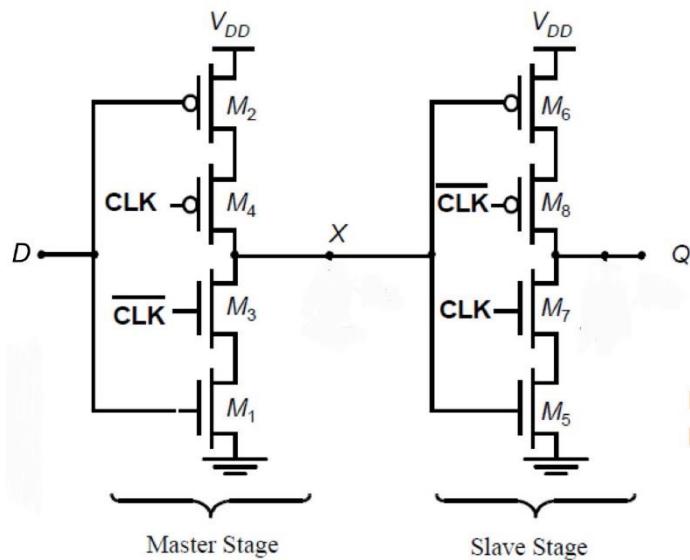
flip-flop 實作上可以有許多架構都能實現 flip-flop 的功能，而我們考慮了三種實作的方式：

(1) 普通的 master-slave 架構



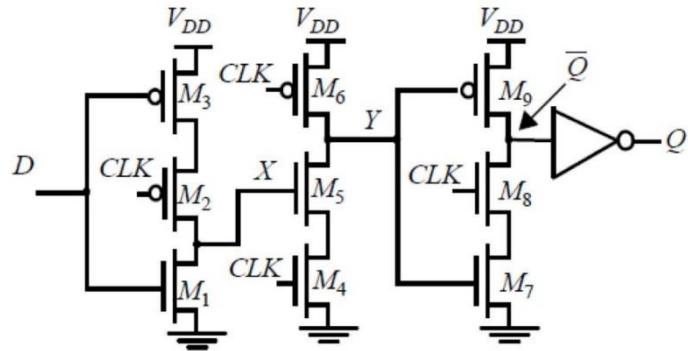
優點是電路的穩定性很高，且後續的 layout 好畫，缺點是電晶體的數量多(18 個)。

(2) C²MOS master-slave 架構



優點是電晶體的數量比起前一種架構少(10 個)，缺點是電路的穩定性很低。

(3) TSPC 架構

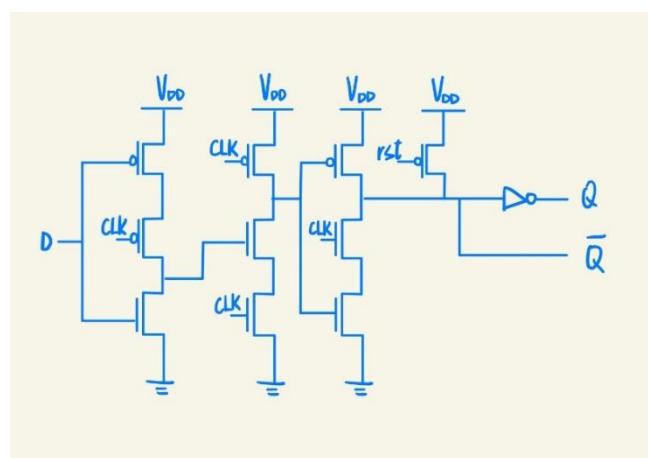


優點是電晶體的數量比起第一種架構少(11 個)，且實測後發現穩定性比C²MOS 還高，缺點是電路的穩定性比第一種架構低一些，且 layout 較難畫。

綜合我們想要將面積壓低(減少電晶體數目)的目標，以及電路穩定度的考量，最終我們選擇使用 TSPC 架構來作為 flip-flop 使用。

決定了 flip-flop 架構後，由於前面提到我們的 flip-flop 要能有 rst 功能，並且能同時輸出正反訊號，因此我們還得做一些變化來達成這兩個條件，從原本 TSPC 的架構可以看到其實 TSPC 本身就有~Q 的訊號了，只是這個訊號相較於 Q 會有些不穩定，且因為兩個訊號之間經過了一個 inv，因此 Q 與~Q 會有一些 delay，但我們為了面積考量，因此~Q 訊號還是選擇從 TSPC 中，M9 與 M8 之間的訊號取得。

而 rst 的實作，由於 SPEC 講義上給的圖是 negative-trigger 的 rst，因此我們決定實作 negative-trigger 的 rst，實作方式則與 decoder 中 enable 的實作方式相同，透過給入一顆連接在~Q 的 pMOS，當 rst 等於 1 時，pMOS 會關閉，此時會輸出正常的 Q 與~Q；而當 rst 為 0 時，pMOS 會開啟，此時會將~Q 的電位藉由 VDD 充電成 1，inverter 出來後的 Q 就會被強制拉成 0，來達成讓 Q 被 rst 成 0，~Q 被 rst 成 1 的功能，並且一旦 rst 給出 0 的訊號時，pMOS 就會開啟，不會等到下一個 clk 才作用，因此這也是 asynchronous 的 rst，最後的 TSPC 架構如下圖。



有了 flip-flop 後，就能來設計除頻電路，而除頻電路又分為同步與非同步，同步的優點是所有頻率的 output 的 delay 相對較小，缺點是同步電路需要更多的電晶體數量去做邏輯的判斷，而非同步的優點是電晶體的數量少，缺點是 output 的 delay 較大且難以控制，若是讓 frequency divider 的 output delay 變大的話，decoder 收到的 input 訊號就會較不整齊，容易產生 glitch 或是 bit line 無法充電到 1.8V 的問題，甚至難以達成各種 SPEC。因此我們最終選擇使用同步的架構，比起單純的壓低面積，我們犧牲一些面積的效益來換取頻率與電路的穩定性。

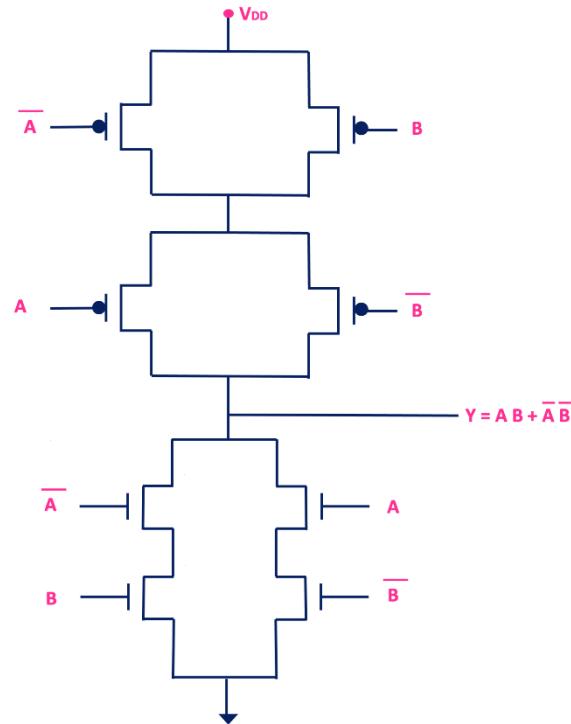
接著是同步電路架構的細節，首先是 2 倍的除頻，我們只需要將 f_2_inv 訊號回授給 TSPC 的 D 端即可，因此之後著重討論 4 倍到 64 倍的同步除頻電路要如何實作。

除頻電路本質上就是 counter，而同步除頻電路的原理就是外接 combinational circuit，算出下一次 counter 應該要是什麼值，再輸入給各個 flip-flop，從下圖可以看到，以三個 bit 的 counter [f_8, f_4, f_2]為例，當 f_2 是 1 的時候，f_4 便會 toggle，當 f_4 跟 f_2 都是 1 的時候，則換成 f_8 會 toggle，因此外接 combinational circuit 的本質就是若比自己還小位數的 bit 都是 1 時，自己就要 toggle。

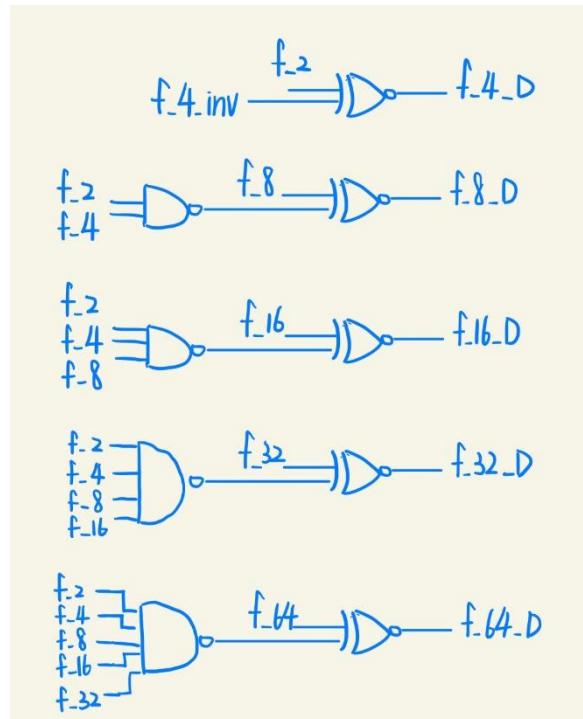
	f_8	f_4	f_2
0	0	0	0
0	0	1	—
0	1	—	0
0	—	1	—
1	—	0	0
—	0	—	1
—	1	—	0
1	1	1	1

上述的邏輯行為傳統上會使用 and 來偵測比自己還小位數的 bit 是否都是 1，並用 xor 來做 toggle，但 and gate 本身需要 6 個電晶體來實作，因此我們將 and 換為用 nand 實作，達到節省電晶體的目的，並且當所有的比自己還小位數的 bit 都是 1 時，nand 會輸出的是 0，因此後面的 toggle 必須改用 xnor 來實作。其中 nand 的電路較簡單，因此著重介紹 xnor 的實作，由於其他方式實作的 xnor 都沒有 cmos 實作的 xnor 來的穩定，因此我們使用 cmos 來實作 xnor，傳統上以 cmos 實作的 xnor 會如下圖。

XNOR Gate

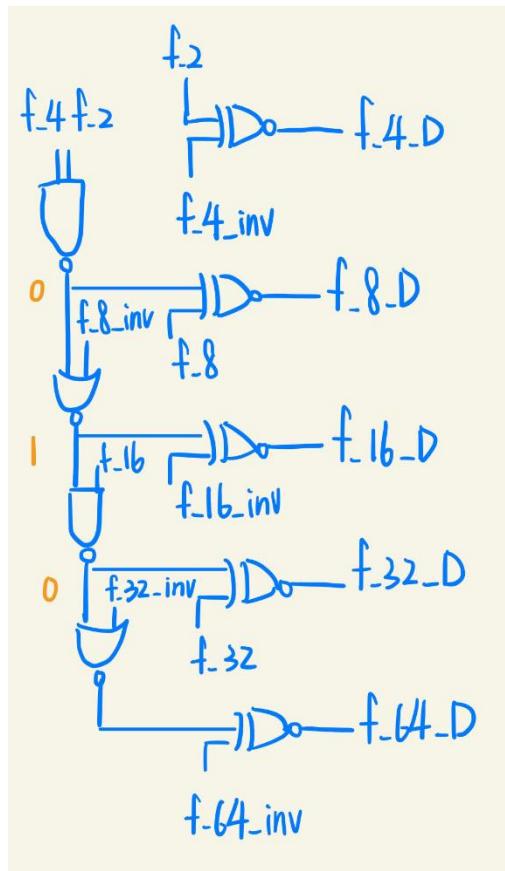


可以看到 XNOR 看似需要 8 個電晶體，但輸入卻需要 A、B 以及 $\sim A$ 、 $\sim B$ ，因此若要做出 2 input 的 XNOR 的話，其實需要 12 個電晶體，但是我們的 TSPC 其實可以給出正反訊號，因此可以讓 XNOR 輸入 A、 $\sim A$ 與 B，這樣電晶體數量就變成 10 個了，設計出來的電路如下圖：



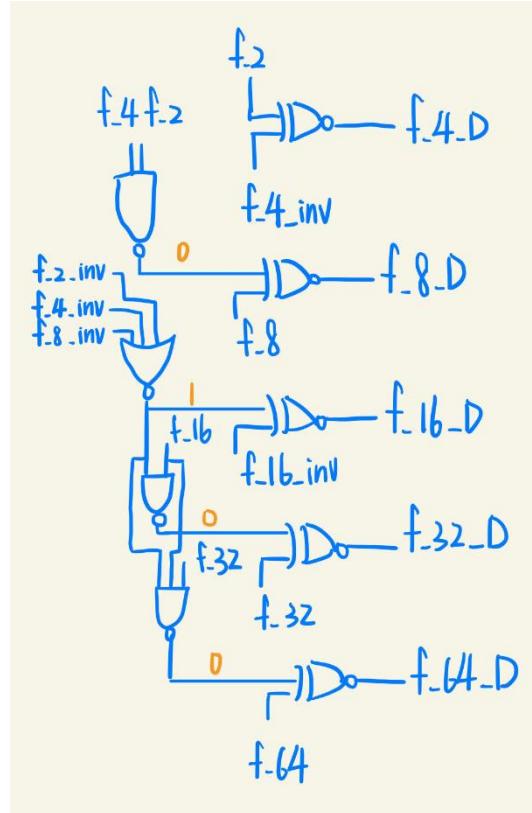
雖然我們的 XNOR 是 3 input 的 gate，但為了方便解釋，圖中所畫的都是 2 input 的 XNOR gate，但其實若其中一個 input 傳入的是 f_2 到 f_8 的正反訊號的話，我們也會傳入對應的反正訊號，因此只畫出 2 個 input。

由於 toggle 行為所採用的 gate 都是 xnor，因此在 f_{4_D} 處需要傳入的訊號會是 f_{4_inv} ，而這種電路的優點就是所有訊號都只經過 nand 與 xnor 兩個 stage 的 delay，缺點就是 nand 有很多 gate 都是超過 2 個 input 的，導致電晶體用量大，因此我們考慮想把所有 nand gate 都換成只用 4 個電晶體的 2 input 的 gate，並且善用我們的 TSPC 能同時給出正反訊號的特性，最後我們想出了下圖的架構。



圖中的黃字表示對應訊號需要 toggle 時，該處的訊號為 1 還是 0，首先是 f_{8_D} 的 stage，當 f_2 與 f_4 都是 1 時表示 f_8 需要 toggle，而 nand gate 則會輸出 0，因此 xnor gate 看到 nand 所輸出的 0 與 f_8 訊號後，就會變成 f_8 訊號的 toggle，而接著看到 f_{16_D} 的 stage，當 f_2 、 f_4 與 f_8 都是 1 時，表示 f_{16} 需要 toggle，而前面當 f_4 與 f_2 都是 1 時，nand gate 會輸出 0，此時我們再利用 nor gate，去判斷 f_{8_inv} 與 nand gate 輸出是否都是 0，若是的話 nor gate 就會輸出 1，同時也代表 f_2 、 f_4 與 f_8 都是 1，而由於 nor gate 輸出 1，因此 xnor gate 要輸入 f_{16_inv} 才能做到 f_{16} 訊號的 toggle， f_{32_D} 與 f_{64_D} 的 stage 也以此類推。

因此我們將本來需要多個 input 的同步電路便成了都是 2 input 的電路，但這樣做有個明顯的缺點，就是對於 f_{64_D} 而言，需要經過 2 個 nand gate、2 個 nor gate 與 1 個 xnor gate，會導致 delay 太大，而使 frequency divider 的頻率不能太高，因此最後我們折衷使用第一個版本與第二個版本的同步電路，得到下圖的架構。



此電路的概念與前一種版本類似，因此不多贅述其功能的實現，可以看到我們將某些 stage 改成多 input 的 gate，這樣雖然會增加電晶體的用量，但每個訊號只會經過最多兩個 nand 或 nor stage 跟一個 xnor stage，大大減少了上一版本 f_{64_D} 的 delay 問題，讓電路的頻率可以變得比較高，最終同步電路決定使用此版本。

3. 各部件尺寸設計

(1) and3_en

首先考慮是否要設計兩種不同尺寸的 and3_en 分別給 bit line、word line 使用，由於兩者 SPEC 的關係互相牽制，同時又要考慮不能太犧牲頻率，因此這部分的設計我們打算先令兩者相同，再依 pre-sim 結果，去決定是否要設計不同的尺寸。

在 and3_en 中，我們總共需要設計的有三部分，第一部分是 nand3 的尺寸，這部分影響的會是 and3_en 訊號輸出與輸入之間的平移，舉例來說，若設計 pMOS 大、nMOS 小，此時會使得 nand3 的 V_m 往 VDD 偏移，代表當輸入訊號是 falling 時，nand3 會有更快的反應速度，表現在 and3_en 最終輸出會下降的比較早；反之當輸入訊號是 rising 時，nand3 會有更慢的反應速度，表現在 and3_en 最終輸出會上升的比較晚。

換句話說，透過增加 nand3 的 pMOS/nMOS ratio，可以使得 and3_en 最終上升的比較晚，下降的比較早，此點能夠解決 SPEC 中 bit line overlapping 不能超過 0.9 V 的問題，但缺點就是因為 bit line 升的比較慢，最終 decoder 能支援的頻率就會因此下降。經過幾次試跑 Hspice 測試的結果，我們發現當 pMOS/nMOS ratio 為 2 時，bit line overlapping 會為 0.2 左右，此時能兼顧 overlapping 與頻率的需求，同時能預防 post-sim 會因為 delay 原因造成 overlapping 上升。尺寸的部分為了有更小的面積，我們最終選用 pMOS 寬為 1 um、nMOS 寬為 0.47 um 的 nand3 設計。

第二部分是 enable pMOS 的尺寸，考慮 enable 純入 0 的時候，此時 pMOS 打開，會將中間連接部分充電為 VDD，但若前述提到的 nand3 此時 pull down 部分也是打開的，這會導致一邊放電另一邊充電的情況發生，因此 enable pMOS 的尺寸要強過於 nand3 pull down 的部分一些，才能夠保證 enable 功能的運作。

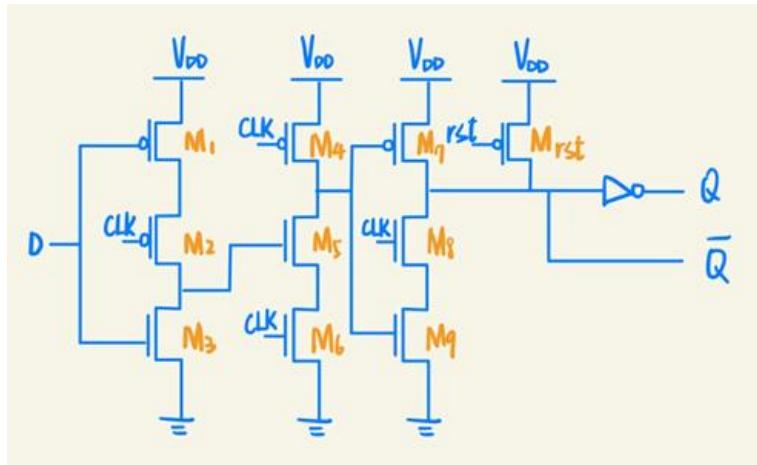
實際測試後我們發現 enable pMOS 的寬至少需要 1.2 um 才能夠在 Waveview 圖形上看到平穩的 enable 結果，因此我們最終選用 pMOS 寬為 1.2 um 的設計。

最後是最後一級 inverter 的尺寸設計，引用 logic effort 的概念，可以推測到最後這級 inverter 尺寸會非常的大。回想前述提到 nand3 是改變輸出上升、下降的早晚(平移)，而在這邊 inverter 就扮演了輸出上升、下降的速度(斜率)，因此也是決定 decoder 頻率的關鍵。

在尺寸的設計上，我們考慮了非常多不同的面向，為了面積與頻率兼顧，以 layout 的角度出發，我們認為既然 enable pMOS 寬度為 1.2 um，那麼 inverter pMOS 寬度若也為 1.2 um 的整數倍，就可以透過 finger layout 的畫法降低空間浪費。因此最終我們選用 pMOS 寬為 4.8 um、nMOS 寬為 3.2 um 的 inverter 設計。

綜上所述，結合 nand3、enable、inverter，我們完成了 and3_en 的設計。

(2) TSPC



TSPC 的寬度較難設計，我們首先看到 TSPC 最後一級其實是個普通的 inverter，我們假設 unskewed inverter 的理想 pMOS/nMOS ratio 為 2，因此我們將該 inverter 的 pMOS/nMOS ratio 先設為 2/1，接著我們先分析 TSPC 的運作原理，TSPC 作為一種 dynamic register，其特點是運作的頻率不可以太低，否則邏輯可能會出錯，首先是當 clk 為 0 時，此時 M2、M4 打開，M6、M8 關閉，此時第一級就如同一個 inverter，而其工作是 sample D 的訊號，讓 M2、M3 之間節點的電壓變為~D，而第二級由於 M4 打開、M6 關閉，因此其工作是 precharge M4 與 M5 之間的節點電壓，使其變為 1，而最後一級的工作是要維持~Q 的訊號，因為 M4 與 M5 之間的節點電壓被 precharge 成 1，因此 M7 關閉，而 M8 也關閉，因此~Q 會被維持住，但此時~Q 的電壓是 floating 的，因此 clk 的速度不可以太慢，否則當~Q=1 時，其電壓可能會逐漸下降。

接著分析 clk 從 0 變為 1 時的運作方式，此時 M2、M4 關閉，M6、M8 打開，這時候第一級的工作仍然是 sample D 的訊號，但由於 M2 關閉，因此會產生一些變化，當 D=1 時，此時 M3 開啟，因此 M2、M3 之間的節點能較為穩定的維持為 0，但當 D=0 時，此時 M3 關閉，M2、M3 之間的節點會是 floating 的 1，而第二級的工作則是計算 M2、M3 之間的節點電壓，並將其反向輸出給 M4、M5 之間的節點電壓，這裡與第一級的狀況相同，由於 M4 關閉、M6 開啟，因此當 M2、M3 之間的節點電壓 =1 時，此時 M5 開啟，M4、M5 之間的節點電壓會從先前 precharge 的 1 變為穩定的 0，當 M2、M3 之間的節點電壓=0 時，此時 M5 關閉，M4、M5 之間的節點電壓會是 floating 的 1，而最後一級的工作，由於 M8 開啟，因此是作為一個 inverter，將 M4、M5 之間的節點電壓反向輸出給~Q。

綜合以上的分析，我們認為在 $\text{clk}=0$ 時，最關鍵的部分應是 M4 與 M5 之間的節點電壓要快速的被 precharge，來讓~Q 訊號能夠維持住，因此我們應將 M4 的寬度設計大一點，而另一重點是在很多狀況下的節電電壓都會是 floating 的，並且電壓都是 floating 的 1，因此我們認為應將 Nmos 的寬度設小，來防止漏電流過大，使 floating point 的電壓無法維持住。結合這幾點，並且為了讓畫 layout 時能比較整齊，我們最終決定將除了 Mrst 的 MOS 以外，pMOS/nMOS ratio 都設為 2，並且為了讓面積最小，因此將 pMOS 的寬度設為 0.95um、nMOS 的寬度設為 0.47um。

最後是 Mrst 的尺寸，與 enable 的考量相同，透過實際測試後我們發現 Mrst 的寬至少需要 1.2 um 才能夠在 Waveview 圖形上看到穩定的 rst 結果，因此我們最終選用 Mrst 寬為 1.2 um 的設計。

(3) nand2

nand2 的設計我們首先考慮 pMOS/nMOS ratio 的關係，我們假設 unskewed inverter 的理想 pMOS/nMOS ratio 為 2，則 unskewed nand2 的 pMOS/nMOS ratio 應為 1/1，而為了讓面積能夠最小化，我們將 pMOS 與 nMOS 的寬度都定為 0.47um。

(4) nand3

nand3 的設計我們也先考慮 pMOS/nMOS ratio 的關係，我們一樣假設 unskewed inverter 的理想 pMOS/nMOS ratio 為 2，則 unskewed nand3 的 pMOS/nMOS ratio 應為 2/3，但為了讓面積能夠最小化，並且匹配 nand2 的寬度，我們仍然讓 pMOS 與 nMOS 的寬度都定為 0.47um，去試試看這樣寬度的 nand3，能否在現在 decoder 所能支援的頻率下維持住同步電路的功能，而嘗試的結果發現仍能維持住同步電路的功能，因此雖然 nand3 這樣的寬度可能會讓 delay 變得較大，但為了面積的效益，我們還是決定將 pMOS 與 nMOS 的寬度都定為 0.47um。

(5) nor3

nor3 的設計我們也參照前面 nand2 與 nand3 的設計方式，我們一樣假設 unskewed inverter 的理想 pMOS/nMOS ratio 為 2，則 unskewed nor3 的 pMOS/nMOS ratio 應為 6/1，但為了讓面積能夠最小化，並且匹配前兩個 gate 的寬度，我們仍然讓 pMOS 與 nMOS 的 size 都定為 0.47um，去試試看這樣寬度的 nor3，能否在現在 decoder 所能支援的頻率下維持住同步電路的功能，但嘗試的結果很遺憾不能維持住同步電路的功能， f_{64} 的訊號無法產生。

我們推測是 nor3 作為 f_{64_D} 的其中一個 stage，因為後面還連接著 nand3，而我們已經將 nand3 的寬度弄成 skewed 的了，讓其 delay 變得稍大，而 nor3 的 unskewed pMOS/nMOS ratio 本應為 6/1，我們卻讓他變成 1/1，因而導致 f_{64_D} 訊號路徑上的 delay 太大，因此我們折衷將 nor3 的 pMOS/nMOS ratio 改為 2/1，再度測試看看能不能維持住同步電路的功能，而嘗試的結果發現 pMOS/nMOS ratio 改為 2/1 能維持住同步電路的功能，因此最後決定將 pMOS 的寬度訂為 0.95um、nMOS 的寬度定為 0.47um。

(6) xnor

xnor 的設計我們也參照前面 3 種 cmos gate 的設計方式，我們一樣假設 unskewed inverter 的理想 pMOS/nMOS ratio 為 2，則 unskewed xnor 的 pMOS/nMOS ratio 應為 2/1，但為了讓面積能夠最小化，並且匹配 nand3 與 nand2 gate 的寬度，我們仍然讓 pMOS 與 nMOS 的 size 都定為 0.47um，去試試看這樣寬度的 xnor，能否在現在 decoder 所能支援的頻率下維持住同步電路的功能，而嘗試的結果發現仍能維持住同步電路的功能，因此雖然 xnor 這樣的寬度可能會讓 delay 變得較大，但為了面積的效益，我們還是決定將 pMOS 與 nMOS 的寬度都定為 0.47um。

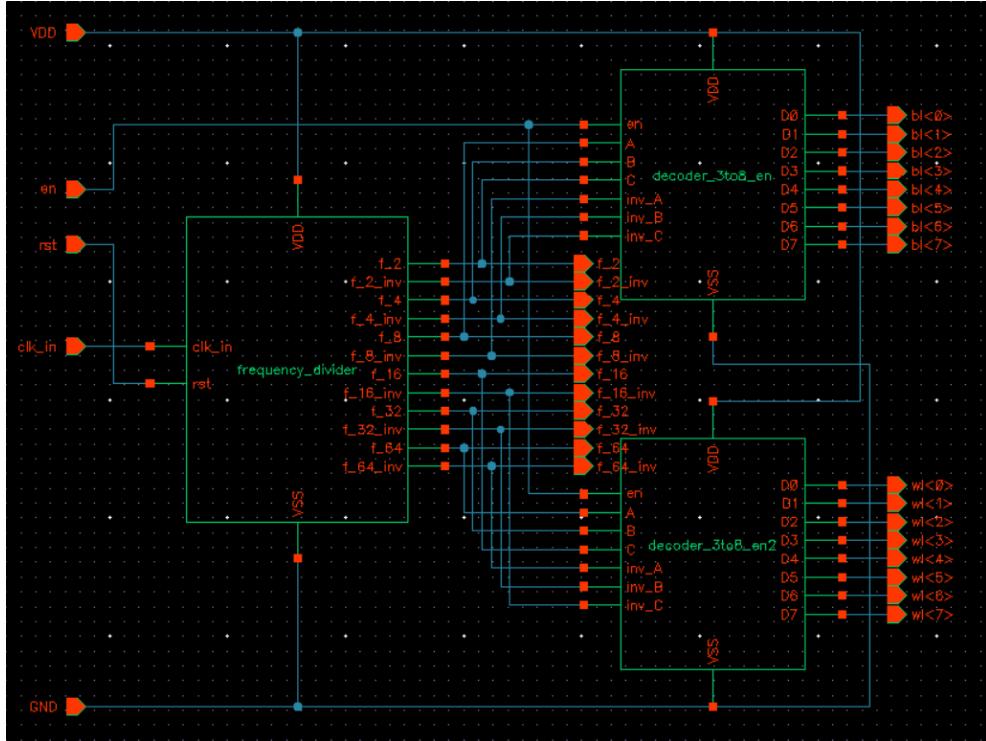
(7) 小結

在設計尺寸的部分中，我們發現 frequency divider 中電路面積能壓多少，是取決於設計的 decoder 能支援的頻率的。舉例來說，若是 decoder 支援的頻率到 500 MHz，那麼在 frequency divider 就可以試著一直降低尺寸到 500 MHz 跑不動為止，這也是我們在設計尺寸上會先去設計 decoder 的原因。

當 frequency divider、decoder 兩者能支援的頻率相當，這時候才不會有任何一邊浪費空間的問題發生，這點也是我們在追求面積之下所體會到的設計重點。

4. Blocks schematic

(1) total

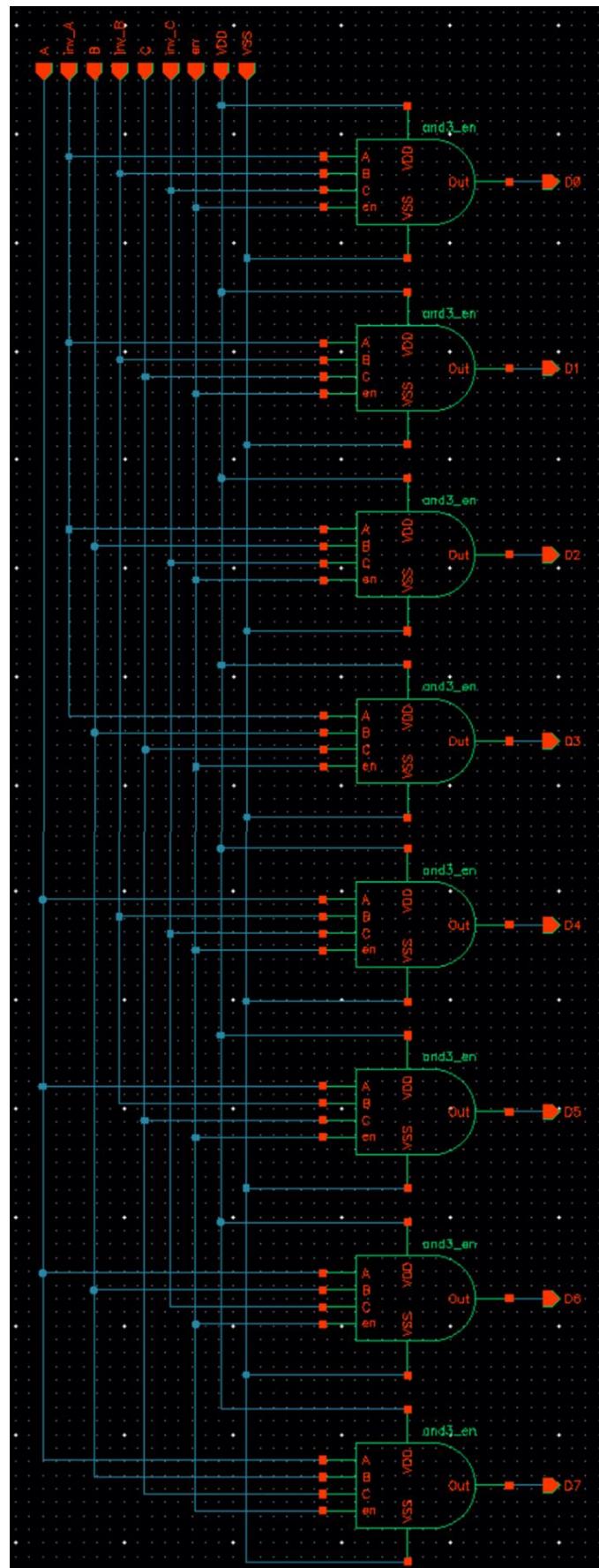


如前述，整個電路由一個 frequency divider 與兩個 decoder 組成，clk_in 會給入 frequency divider 去除頻出 $f_2 \sim f_{64}$ 的正反訊號。

而 frequency divider 需要傳入 $f_2 \sim f_8$ 的正反訊號給入 decoder 1，得到 bit line 的 8 個 output；同時，frequency divider 也需要傳入 $f_{16} \sim f_{32}$ 的正反訊號給入 decoder 2，得到 word line 的 8 個 output。

最後記得 reset 需給入 frequency divider、enable 需給入兩個 decoder，去完成各 blocks 初始化的動作。

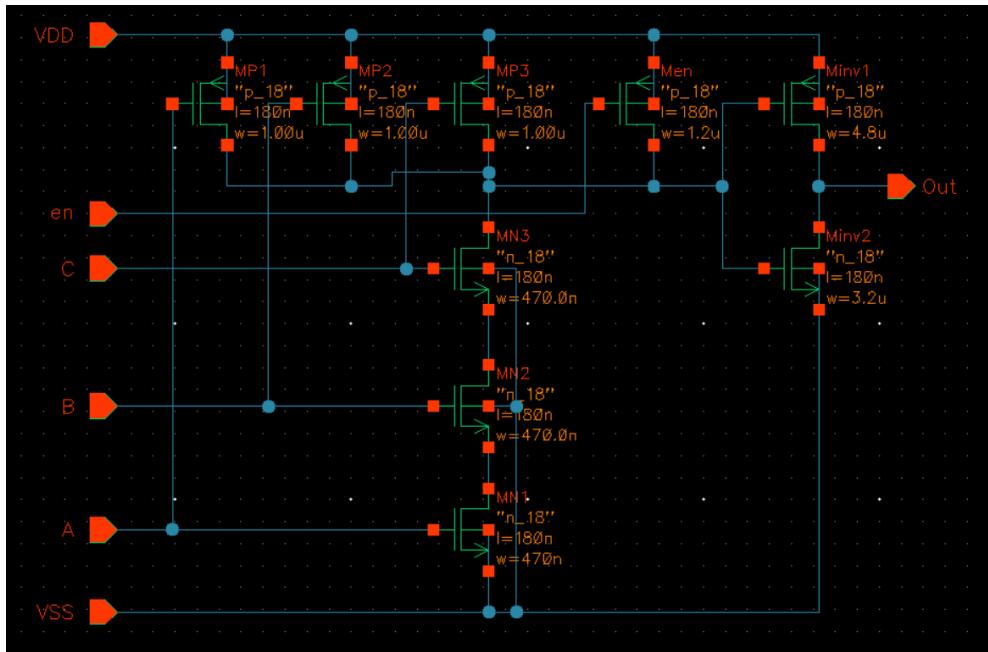
(2) decoder 1 & 2



decoder 1 & 2 的差異只在於使用 and3_en 1 與 and3_en 2，因此可以設計出不同的 size 去有彈性地調控 bit line 與 word line 的電性表現差異，各個輸出與輸入關係如前述：

$$\begin{aligned}
 D0 &= enable \& \sim A \& \sim B \& \sim C \\
 D1 &= enable \& \sim A \& \sim B \& C \\
 D2 &= enable \& \sim A \& B \& \sim C \\
 D3 &= enable \& \sim A \& B \& C \\
 D4 &= enable \& A \& \sim B \& \sim C \\
 D5 &= enable \& A \& \sim B \& C \\
 D6 &= enable \& A \& B \& \sim C \\
 D7 &= enable \& A \& B \& C
 \end{aligned}$$

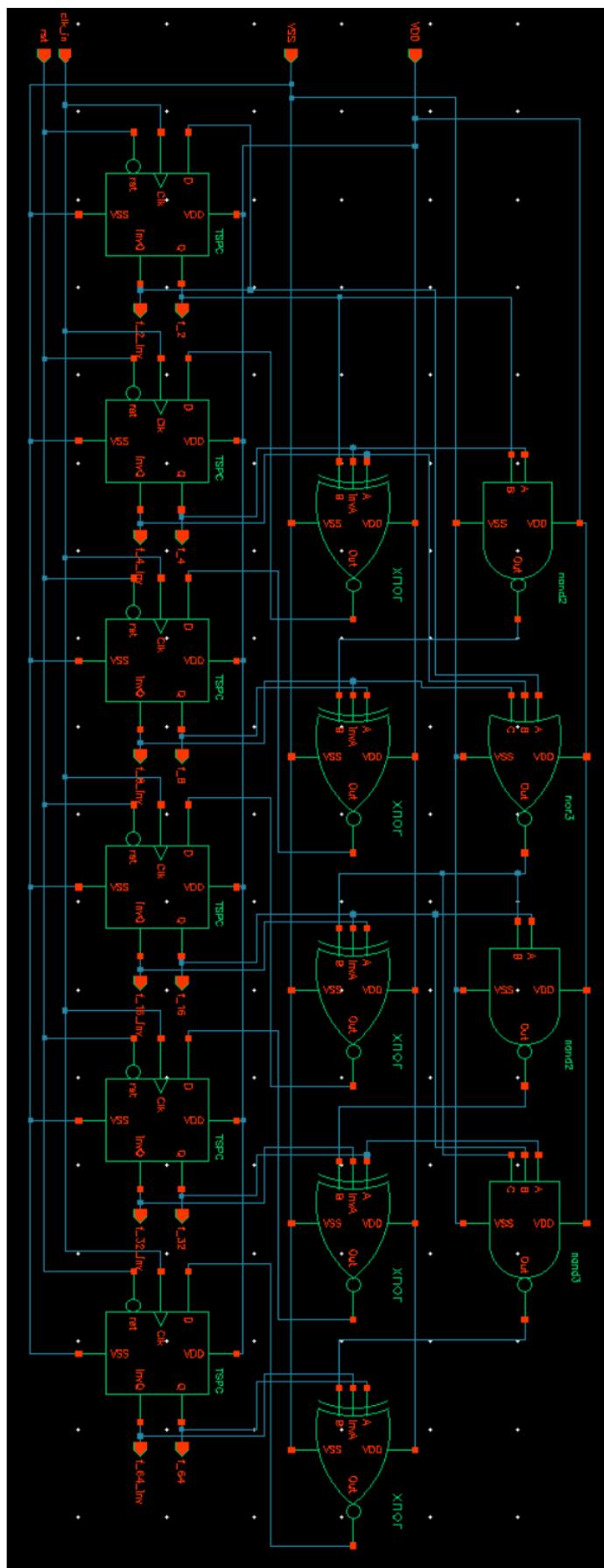
(3) and3_en 1 & 2



如前述，使用左邊 6 顆 MOS 的 nand3 加上右邊 2 顆 MOS 的 inverter 組成主要架構，中間的 pMOS 為 enable 用。

所有 MOS 尺寸皆照前述設計。

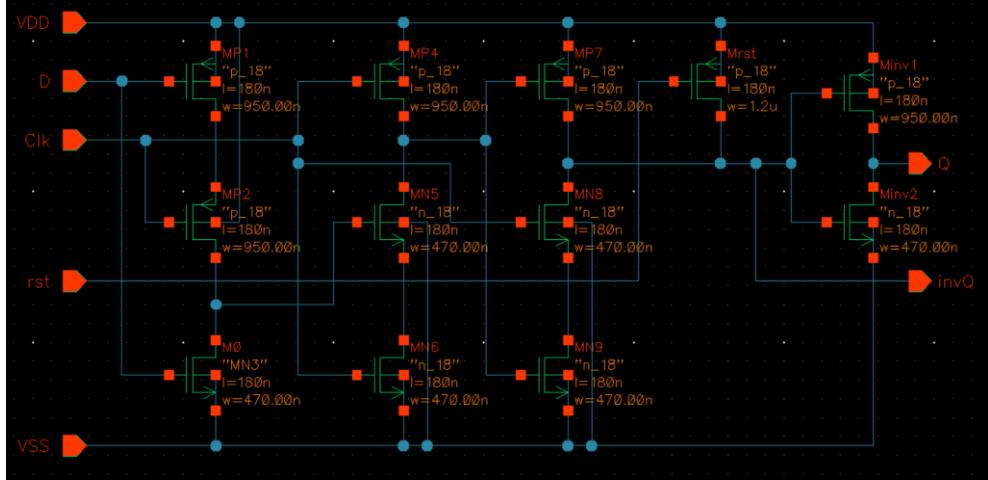
(4) frequency divider



如前述，frequency divider 使用同步化的設計，因此除了除頻用的 6 個 TSPC 外，還使用了許多組合電路去完成同步化的準備工作。

各 blocks 之間的連結如前述與上圖所示，因此不再多做贅述。

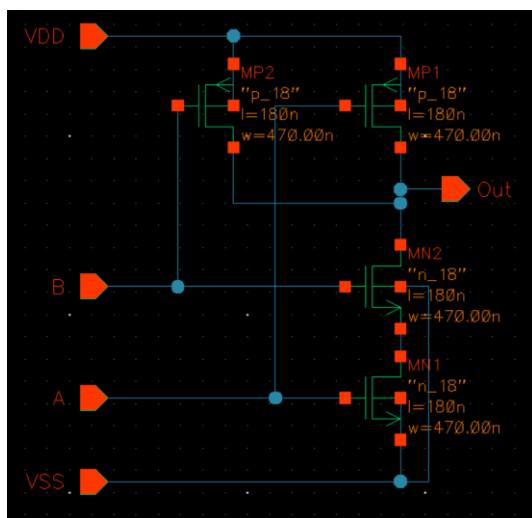
(5) TSPC



如前述，使用左邊 9 顆 MOS 的 TSPC 架構去完成 Flip flop 的主要功能，加上右邊 2 顆 MOS 的 inverter 去將輸出穩定與反轉，中間的 pMOS 為 reset 用。

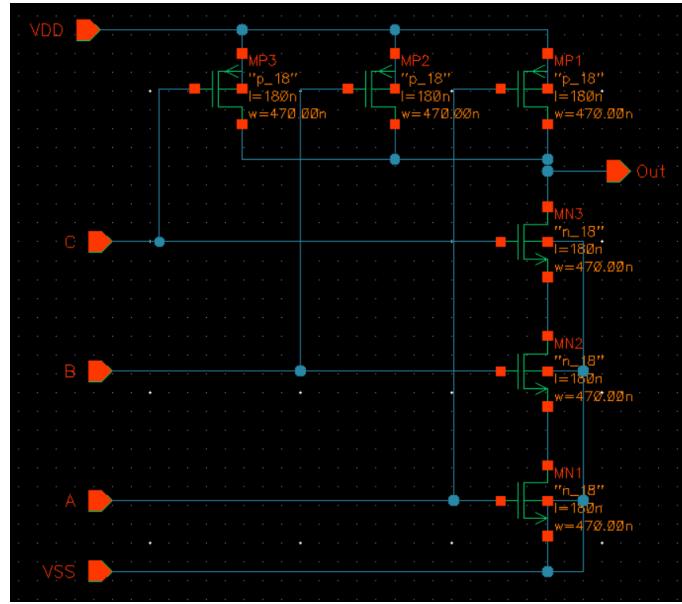
所有 MOS 尺寸皆照前述設計。

(6) nand2



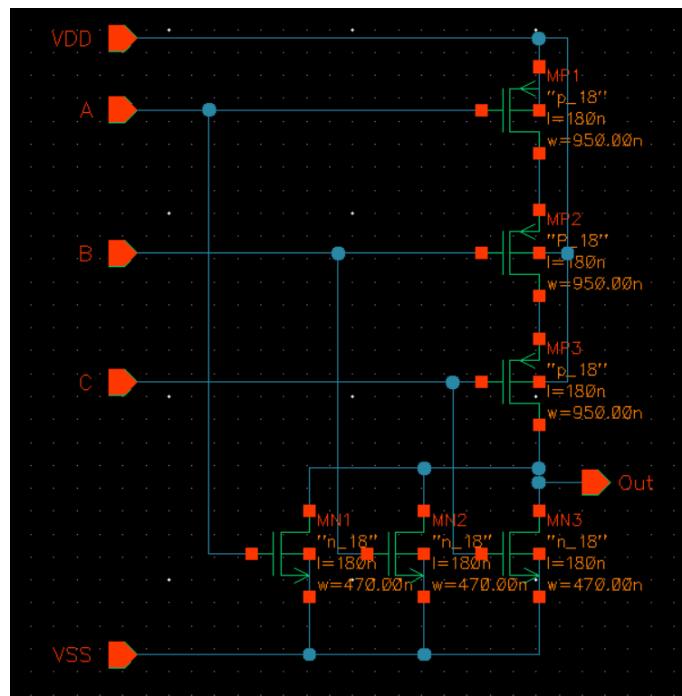
nand2 功能較為簡單，因此不再多做贅述，所有 MOS 尺寸皆照前述設計。

(7) nand3



nand3 功能較為簡單，因此不再多做贅述，所有 MOS 尺寸皆照前述設計。

(8) nor3



nor3 功能較為簡單，因此不再多做贅述，所有 MOS 尺寸皆照前述設計。

5. Pre-sim 結果檢查

(1) 模擬環境

溫度：25 度 pw : 820 ps 頻率：550 MHz

Reset : 30 ~ 40 ns Enable : 50 ns ~

(2) Demo standard

有些 standard 只在某些 corner 難以達成，因此我們會先列出哪些 standard 要在哪個 corner 仔細去看，避免重複一直貼混亂的 Waveview 結果。

- (a) When $BL< n+1 >$ turn on , $BL< n >$ must turn off, intersection of two lines " $V \leq 0.9(V)$ "

⇒ FF corner，因充放電較快，容易 overlapping。

- (b) When $WL< n >$ turn on , $BL< 0 >$ must be ready. ($1.8(V)$)

⇒ SS corner，可以更仔細的看 WL、BL 的升降關係。

- (c) Each BL must overlap with WL for more than $0.1(nsec)$ *note : Both are $1.8 (V)$

⇒ SS corner，因充放電較慢，SS corner 為影響頻率的主因。

- (d) When $BL< 0 >$ turn on , $WL < n-1 >$ must turn off, intersection of two lines " $V \leq 0.4(V)$ "

⇒ FF corner，因充放電較快，容易 overlapping。

- (e) We recommend that $WL< n >$ be designed to shut down earlier than $BL< 7 >$

⇒ SS corner，可以更仔細的看 WL、BL 的升降關係。

- (f) When $WL< n+1 >$ turn on , $BL< 7 >$ must turn off, intersection of two lines " $V \leq 0.4(V)$ "

⇒ FF corner，因充放電較快，容易 overlapping。

- (g) The duty cycle of CLK_in must be 50 %

⇒ 輸入給定時就決定了，不用從結果看

- (h) Glitch must be smaller than $0.9(V)$

⇒ FF corner，因充放電較快，有無 glitch 能明顯看到。

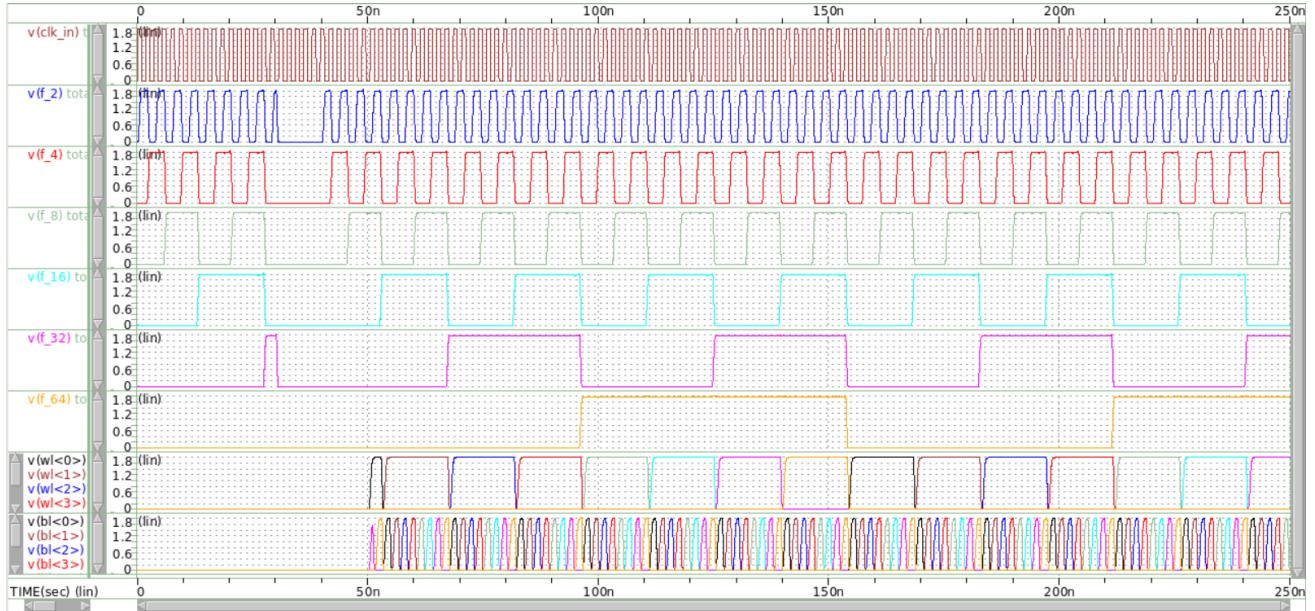
- (i) The changing rate of BL must be determined by f_2

⇒ 任意 corner，可以由結果總圖直接看出。

(3) SS corner

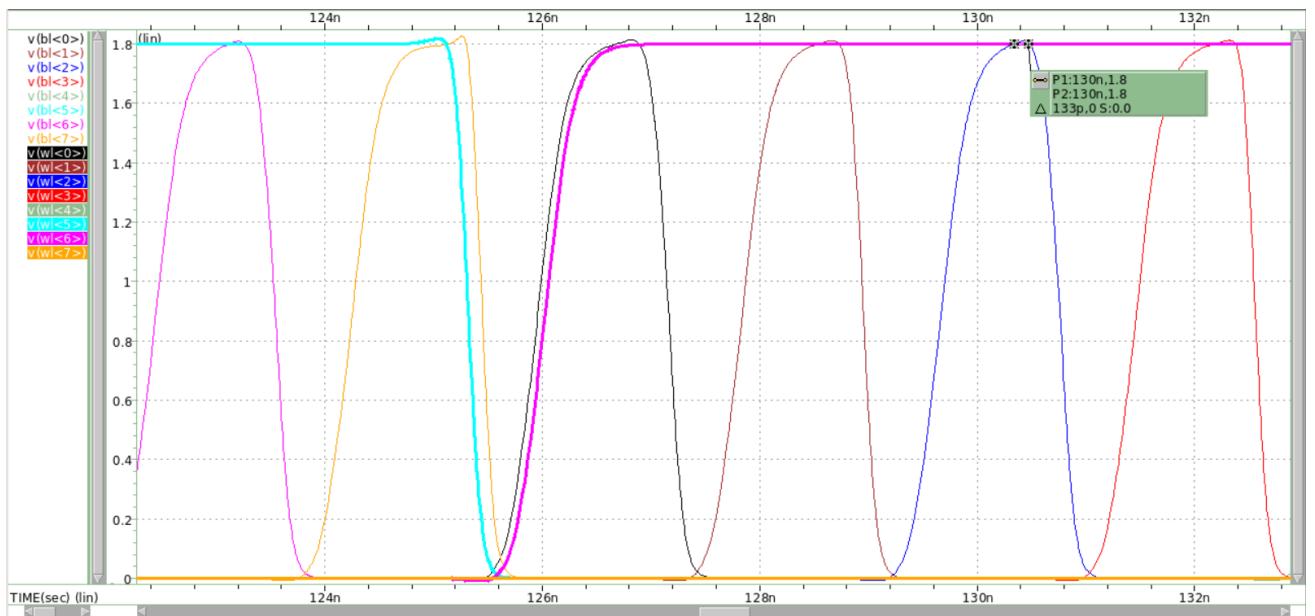
結果總圖：

可以看出 reset、enable、decoder、frequency divider 皆有正常運作。



WL、BL 關係圖：

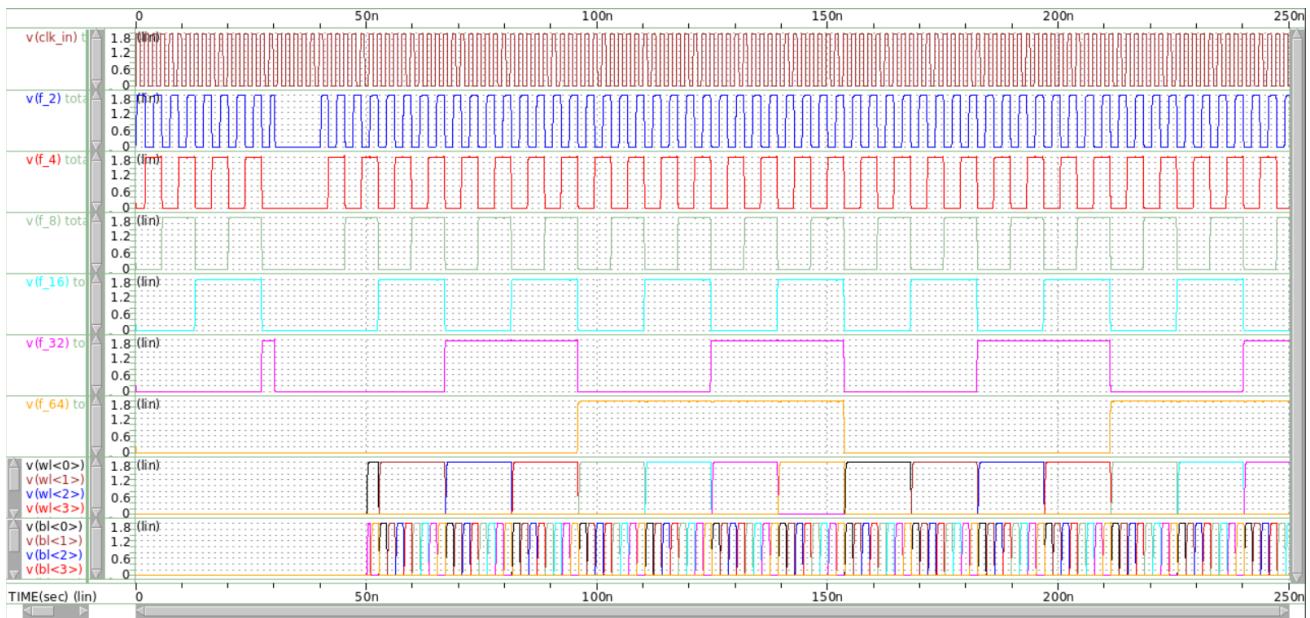
在 SS corner 中，我們著重看 standard (b)(c)(e)，下圖中粗線為 WL、細線為 BL，可以看到在 WL 會比 BL 更早下降、更晚上升，而且每個 BL 與 WL 都有交疊超過 0.1 ns，有滿足 SPEC 所求。



(4) TT corner

結果總圖：

可以看出 reset、enable、decoder、frequency divider 皆有正常運作。



功耗表現：

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
average_power= 472.3290u from= 100.0000n to= 215.4560n
```

取 100 ns 到 215.456 ns 的 64 個 clk_in 週期(一個 f_64 週期)來做功率平均，可以得到功率為 0.47 mW。

(5) FF corner

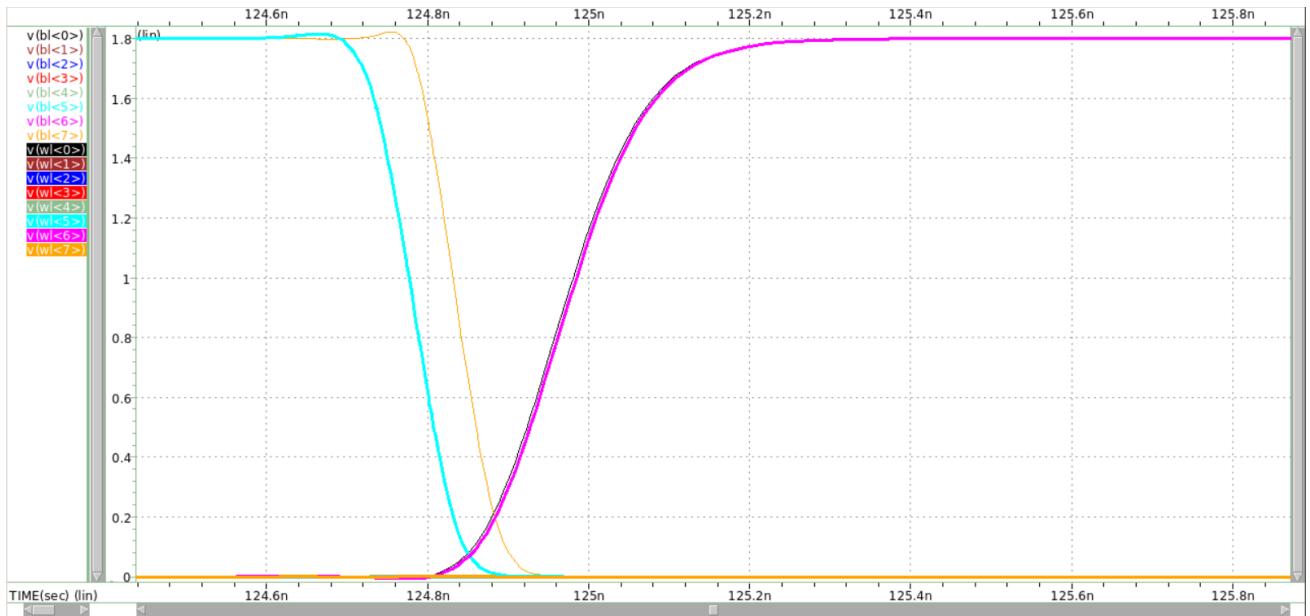
結果總圖：

可以看出 reset、enable、decoder、frequency divider 皆有正常運作。



WL、BL 關係圖：

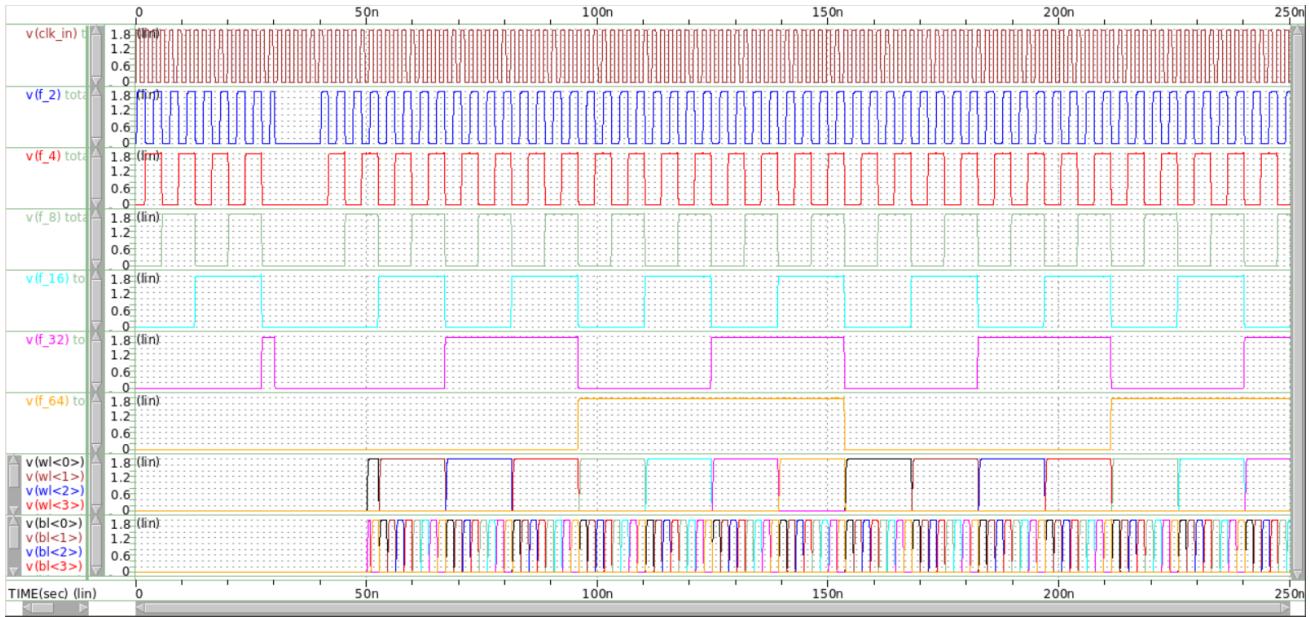
在 FF corner 中，我們著重看 standard (a)(d)(f)(h)，下圖中粗線為 WL、細線為 BL，可以看到在 BL overlapping 在 0.2 V、wl<n>與 bl<0>交在 0.1 V、wl<n+1>與 bl<7>交在 0.2 V，且皆無明顯 glitch 的產生，有滿足 SPEC 所求。



(6) FS corner

結果總圖：

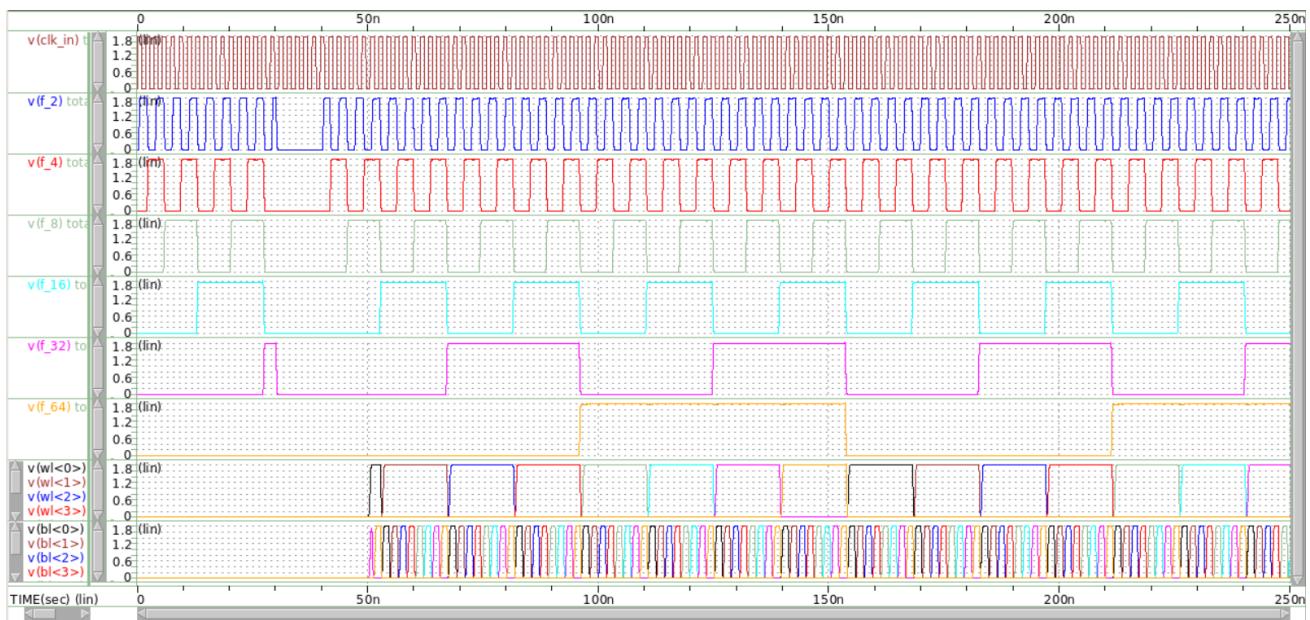
可以看出 reset、enable、decoder、frequency divider 皆有正常運作。



(7) SF corner

結果總圖：

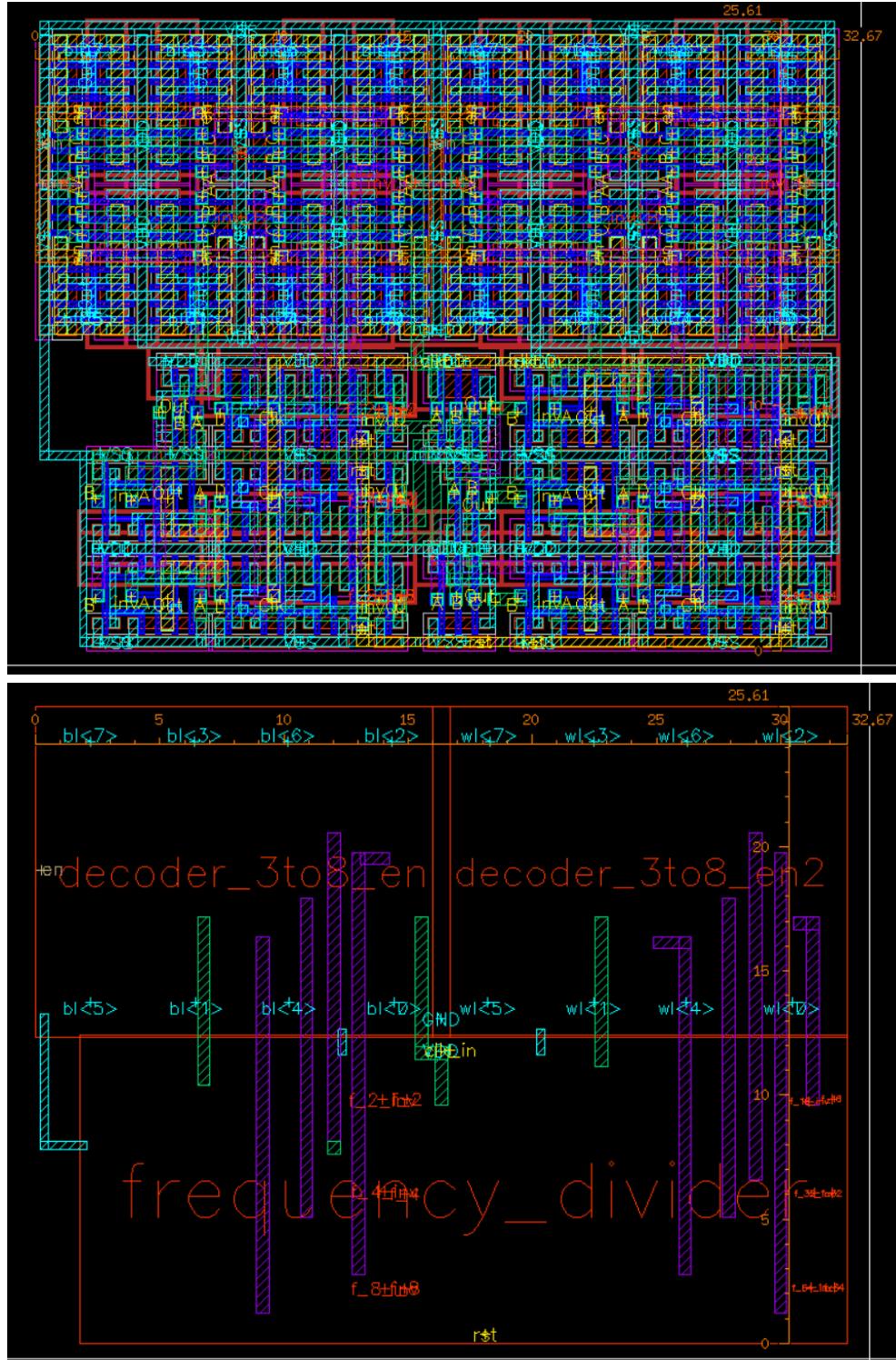
可以看出 reset、enable、decoder、frequency divider 皆有正常運作。



二、Layout

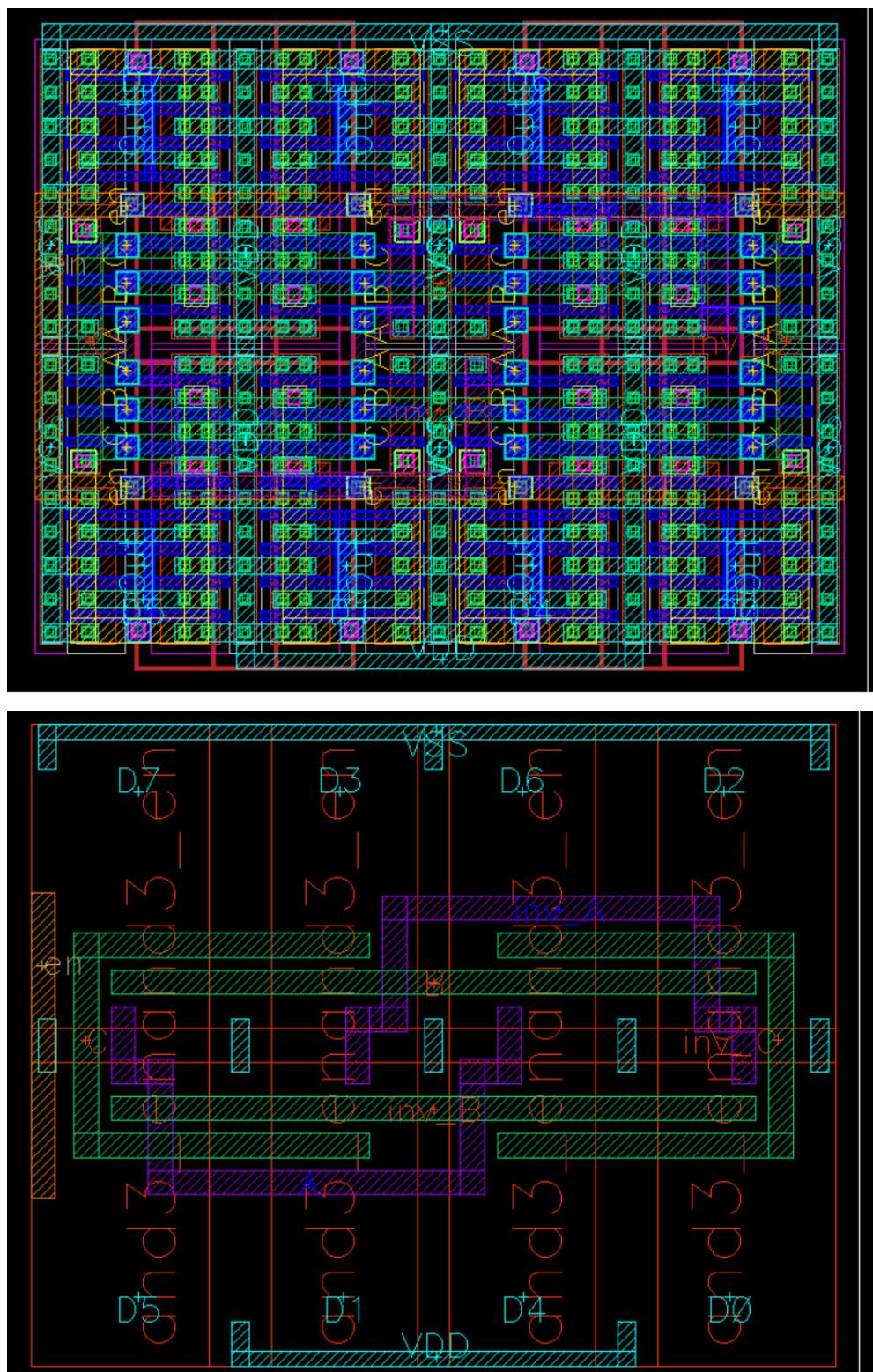
1. Blocks layout

(1) total



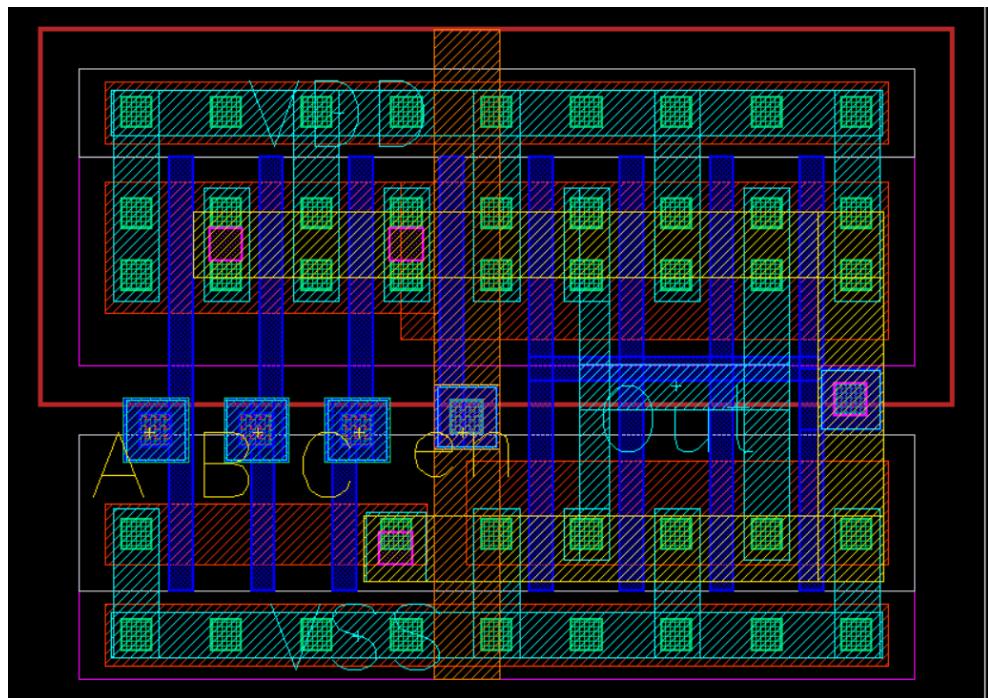
寬 32.67 um、高 25.61 um，總面積為 837 um²。

(2) decoder 1 & 2



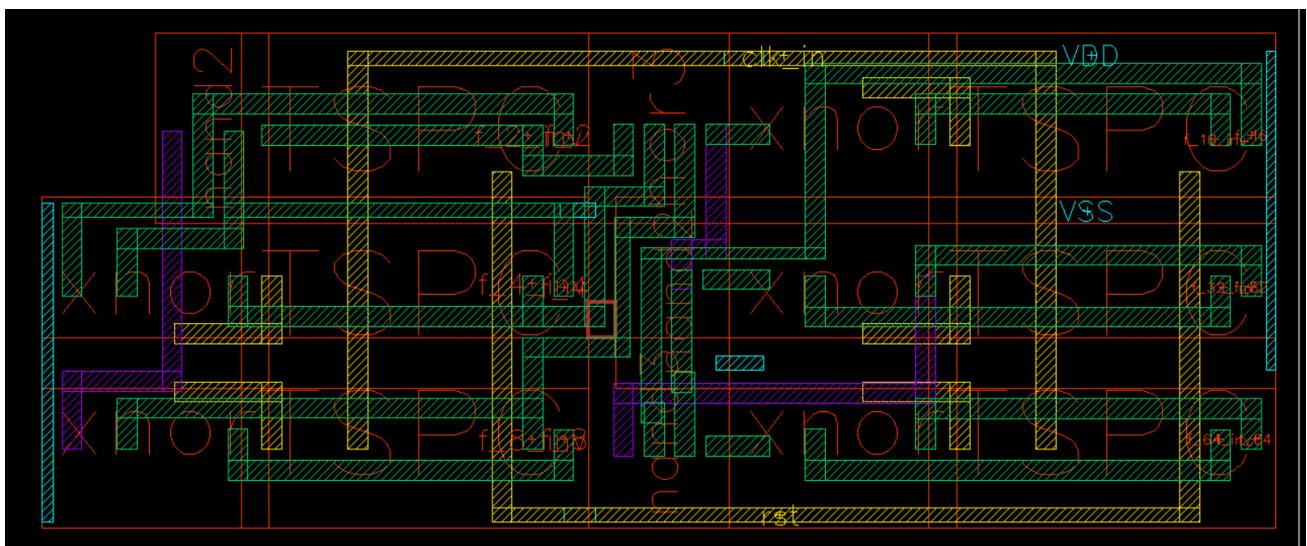
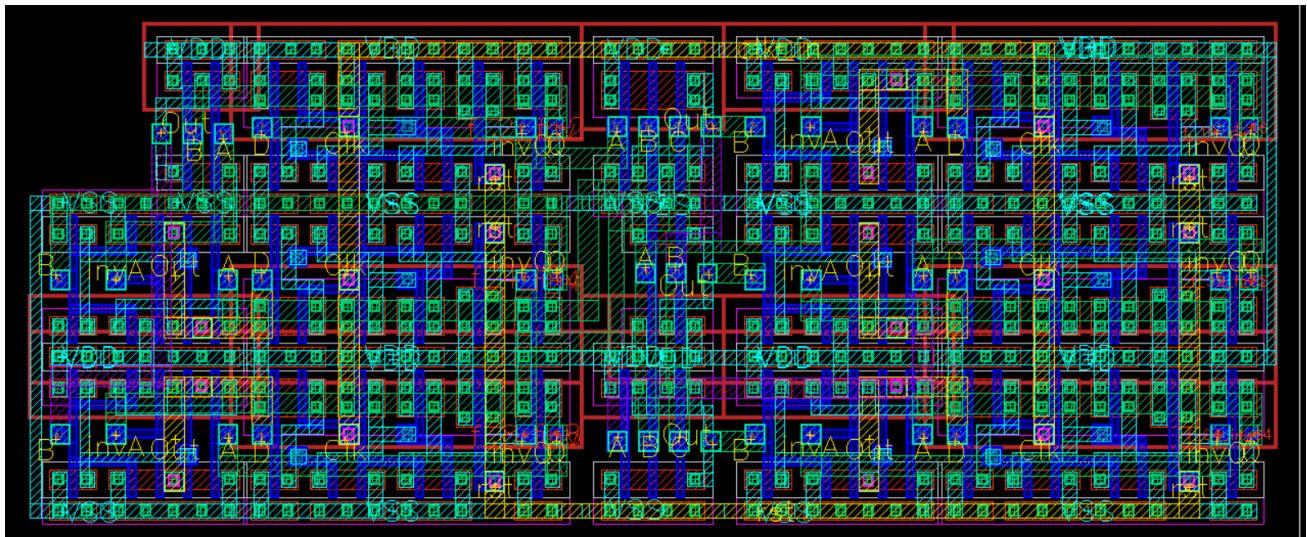
Decoder 1 & 2 雖說使用不同的 and3_en，但兩者尺寸設計皆相同，layout 也畫得一模一樣，因此只附上 decoder 1 的 layout。

(3) and3_en 1 & 2

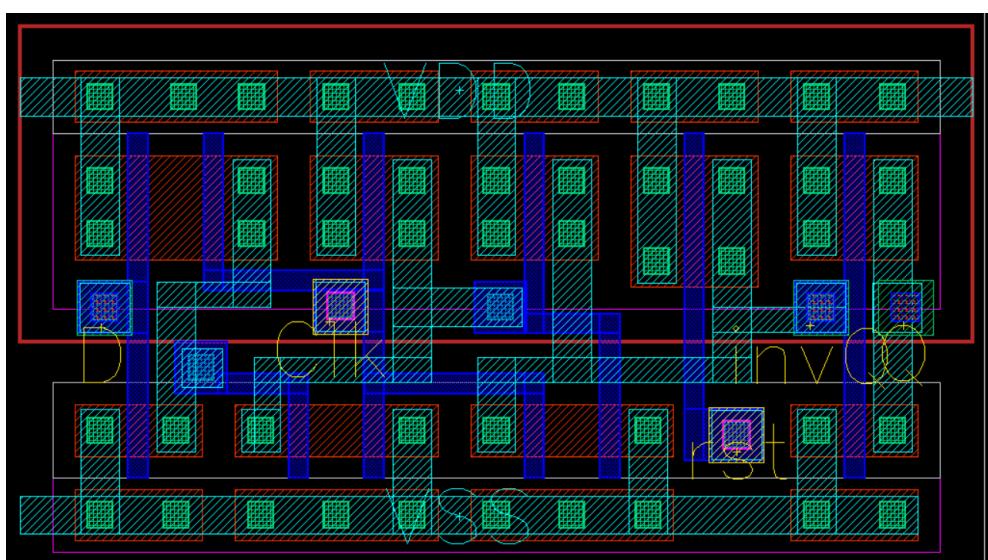


And3_en 1 & 2 尺寸設計皆相同，layout 也畫得一模一樣，因此只附上 and3_en 1 的 layout。

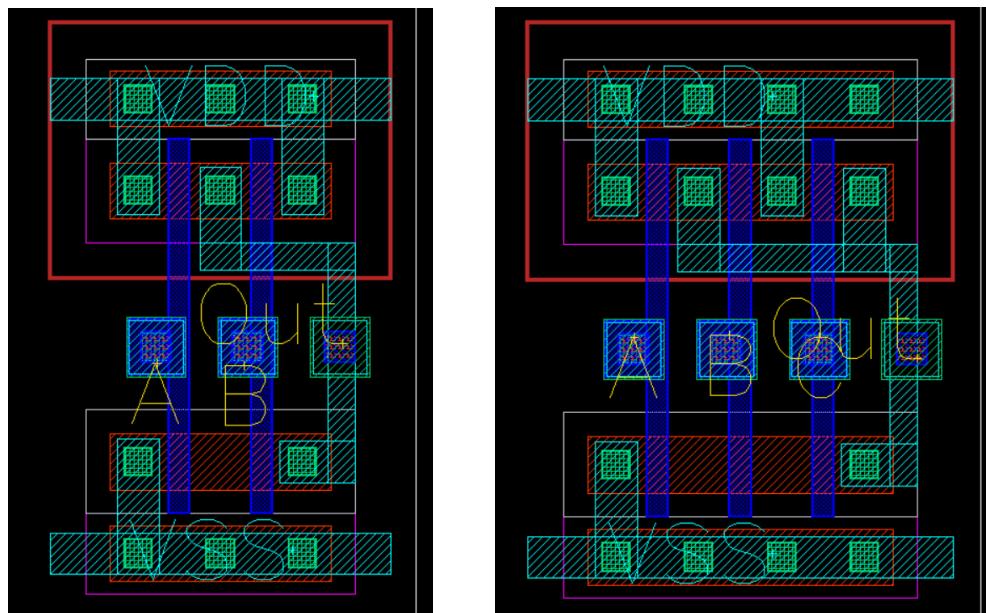
(4) frequency divider



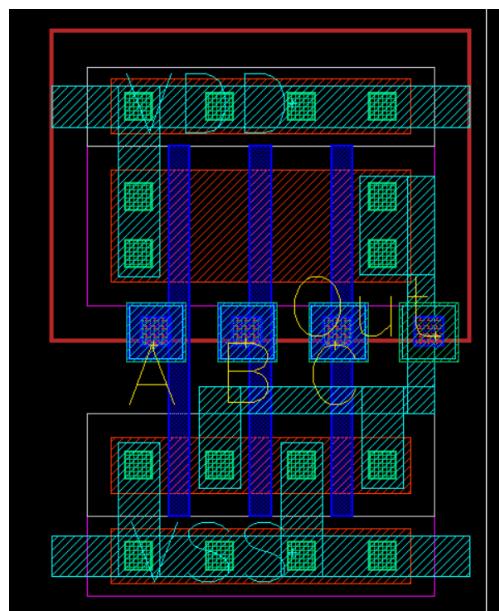
(5) TSPC



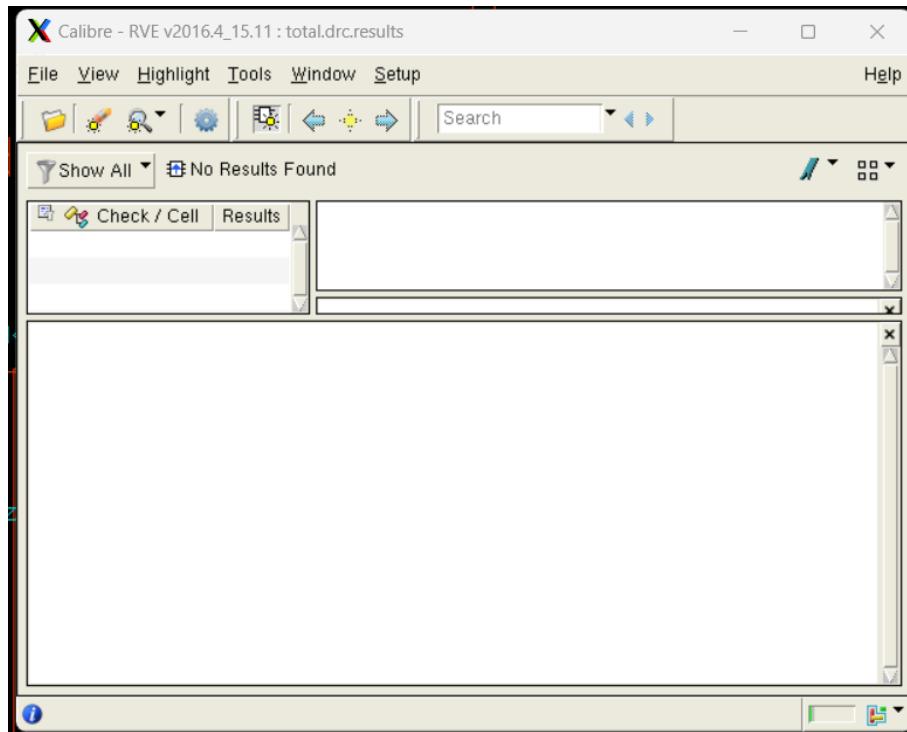
(6) nand2、nand3



(7) nor3

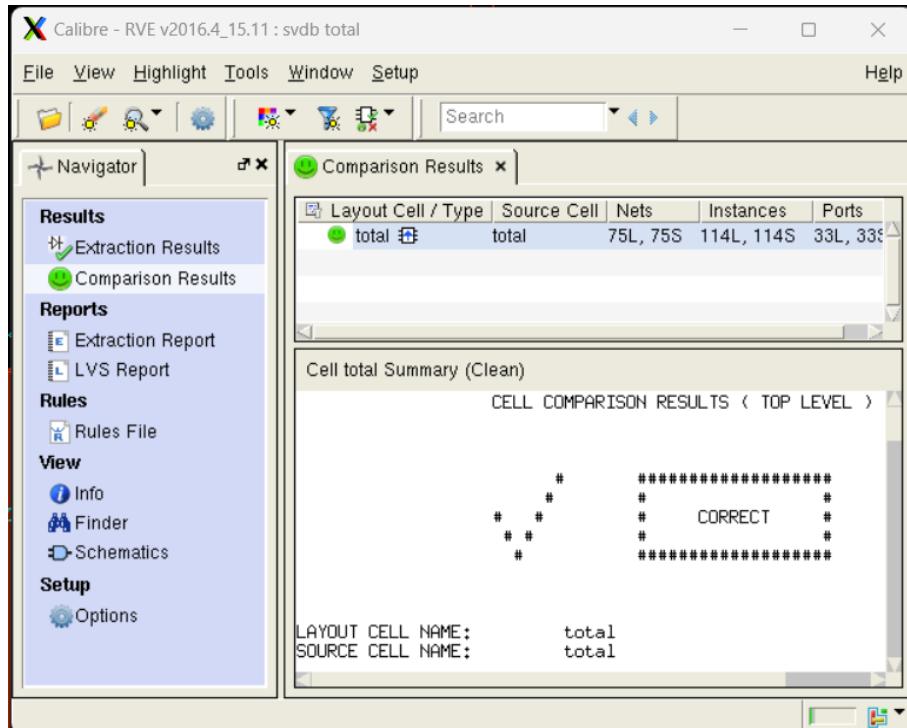


2. DRC 結果



DRC 沒有任何錯誤，皆有滿足 layout 要求。

3. LVS 結果



LVS 沒有任何錯誤，皆有滿足 schematic 的架構與尺寸。

三、Simulation

1. Pre-sim 結果

Pre-sim 的結果在前述已經有放上，因此不再贅述。

2. Post-sim 結果

(1) 模擬環境

溫度：25 度

pw : 970 ps

頻率：470 MHz

Reset : 10~20 ns

Enable : 30 ns ~

```
****Main Circuit****
V1 VDD      GND  1.8
V2 clk_in   GND  pulse  0  1.8    0n  "0.1*pw"  "0.1*pw"  "pw"  "2.2*pw"
V3 rst       GND  pulse  1.8    0  10n    0.01n    0.01n   10n   20u
V4 en        GND  pulse  0  1.8   30n    0.01n    0.01n   320n  20u
```

(2) Demo standard

- (a) When BL_{n+1} turn on , BL_n must turn off, intersection of two lines" $V \leq 0.9(V)$
- (b) When WL_n turn on , BL_0 must be ready. ($1.8(V)$)
- (c) Each BL must overlap with WL for more than 0.1(nsec) *note : Both are 1.8 (V)
- (d) When BL_0 turn on , WL_{n-1} must turn off, intersection of two lines" $V \leq 0.4(V)$
- (e) We recommend that WL_n be designed to shut down earlier than BL_7
- (f) When WL_{n+1} turn on , BL_7 must turn off, intersection of two lines" $V \leq 0.4(V)$
- (g) The duty cycle of CLK_in must be 50 %

⇒ 上圖可以知道我們 clk_in 細的是 1、0 時間相同的，滿足 SPEC。

- (h) Glitch must be smaller than 0.9(V)

⇒ 我們的設計結果皆無 glitch 產生，後續可以由附圖看到。

- (i) The changing rate of BL must be determined by f_2

⇒ 後續可以由附圖看到。

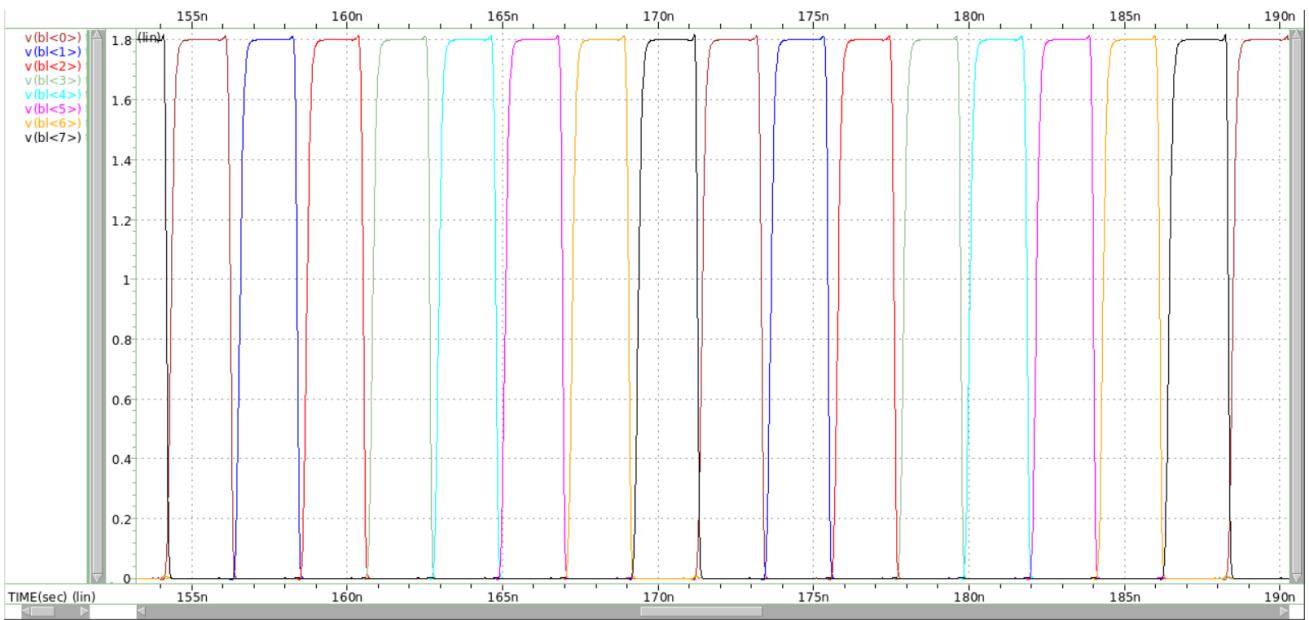
(3) TT corner

結果總圖：

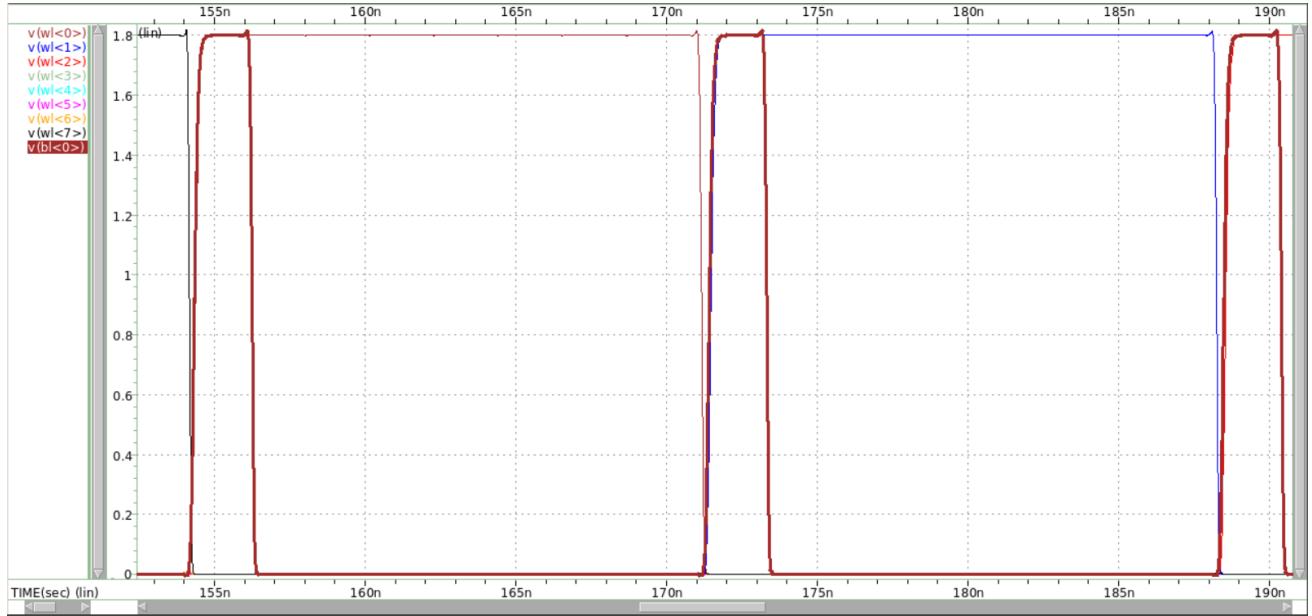
可以看出 reset、enable、decoder、frequency divider 皆有正常運作，此外沒有 glitch 的產生(h)、可以看出 bit line 的改變由 f_2 決定(i)。



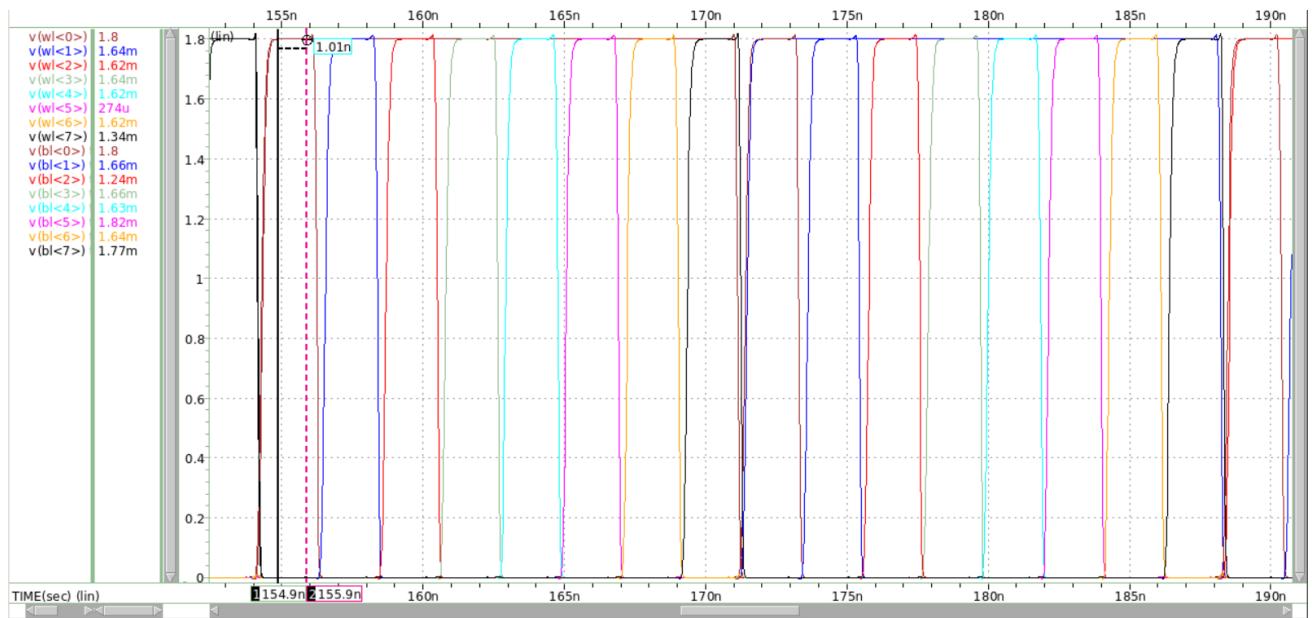
- (a) When $BL< n+1 >$ turn on , $BL< n >$ must turn off , intersection of two lines " $V \leq 0.9(V)$ "



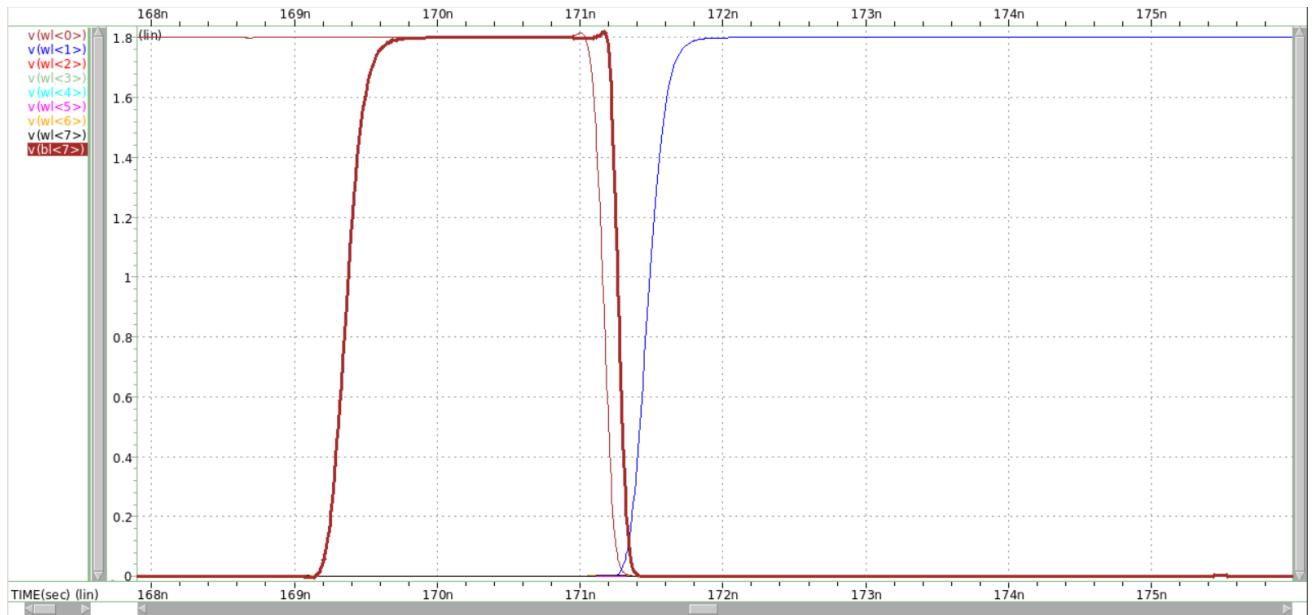
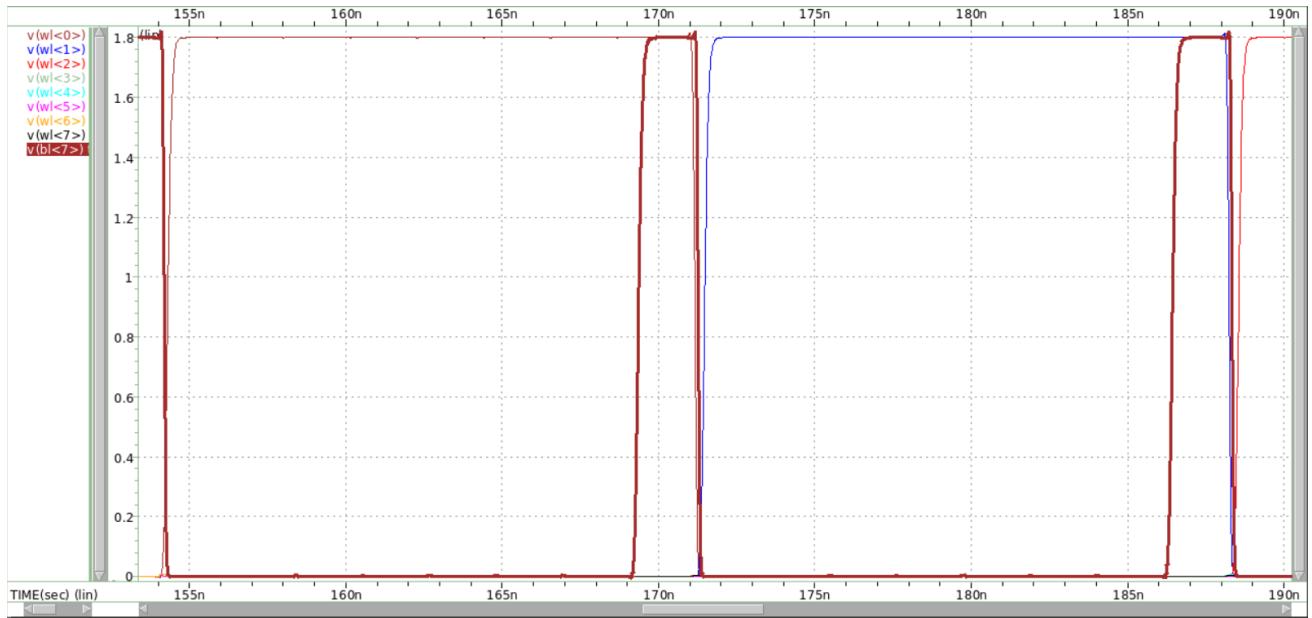
- (b) When $WL<0>$ turn on , $BL<0>$ must be ready. (1.8(V))
- (d) When $BL<0>$ turn on , $WL <n-1>$ must turn off, intersection of two lines " $V \leq 0.4(V)$ "



- (c) Each BL must overlap with WL for more than $0.1(nsec)$ *note : Both are $1.8 (V)$



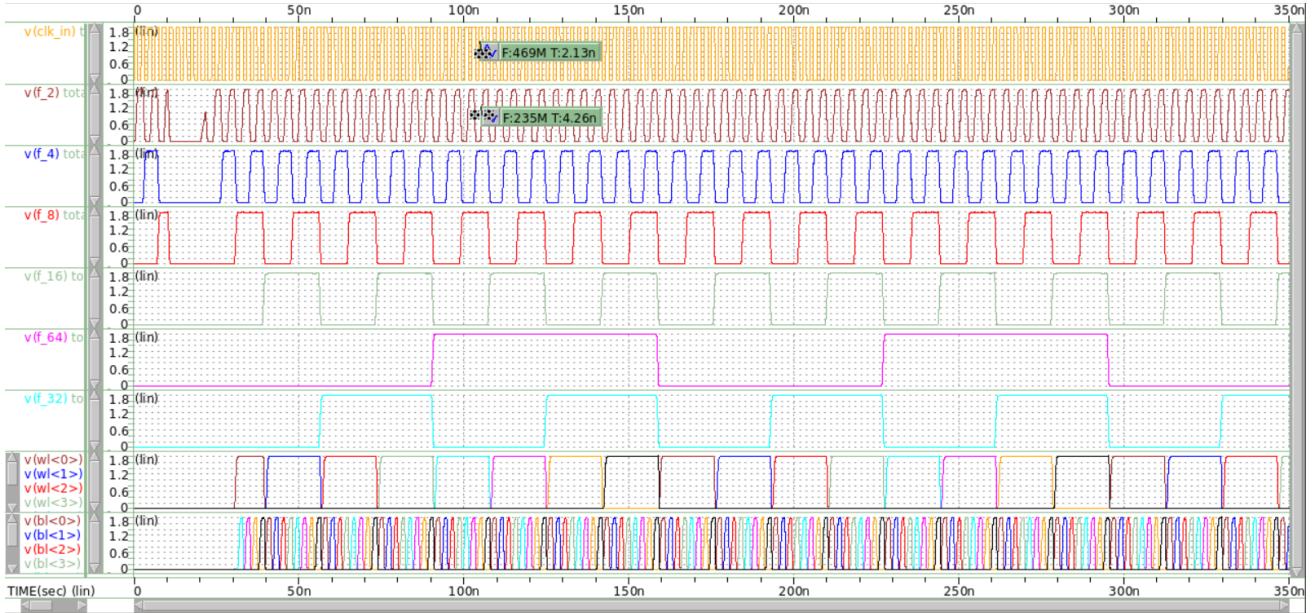
- (e) We recommend that $WL<7>$ be designed to shut down earlier than $BL<7>$
- (f) When $WL<7>$ turn on, $BL<7>$ must turn off, intersection of two lines " $V \leq 0.4(V)$ "



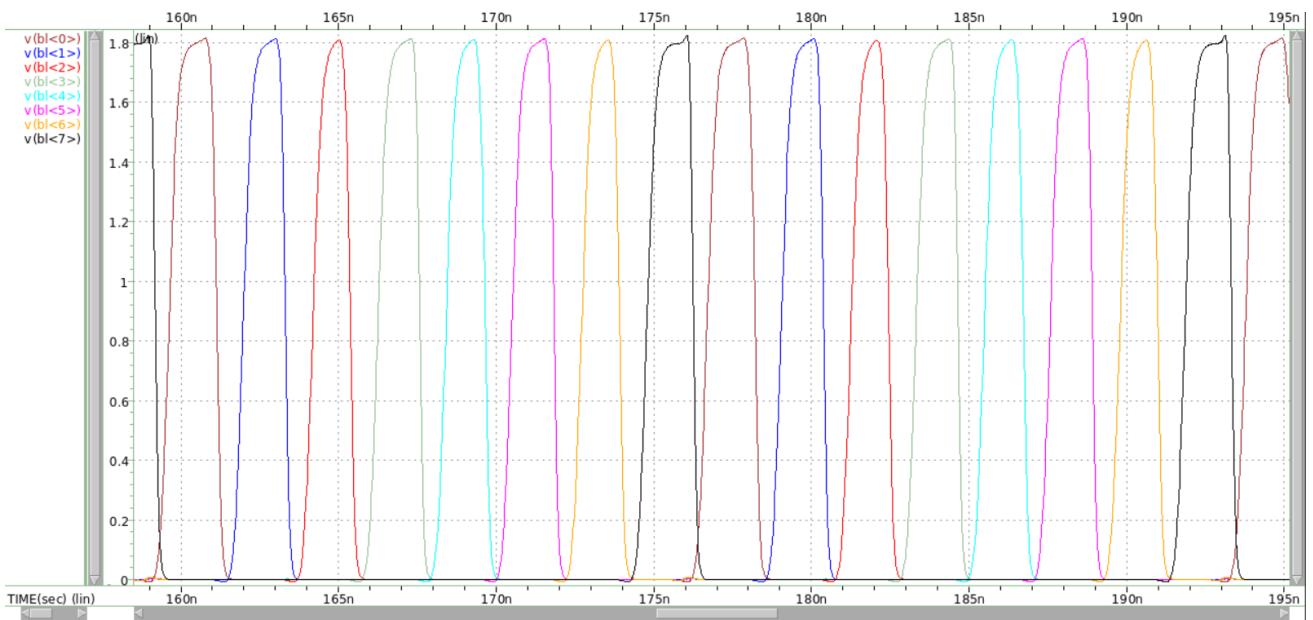
(4) SS corner

結果總圖：

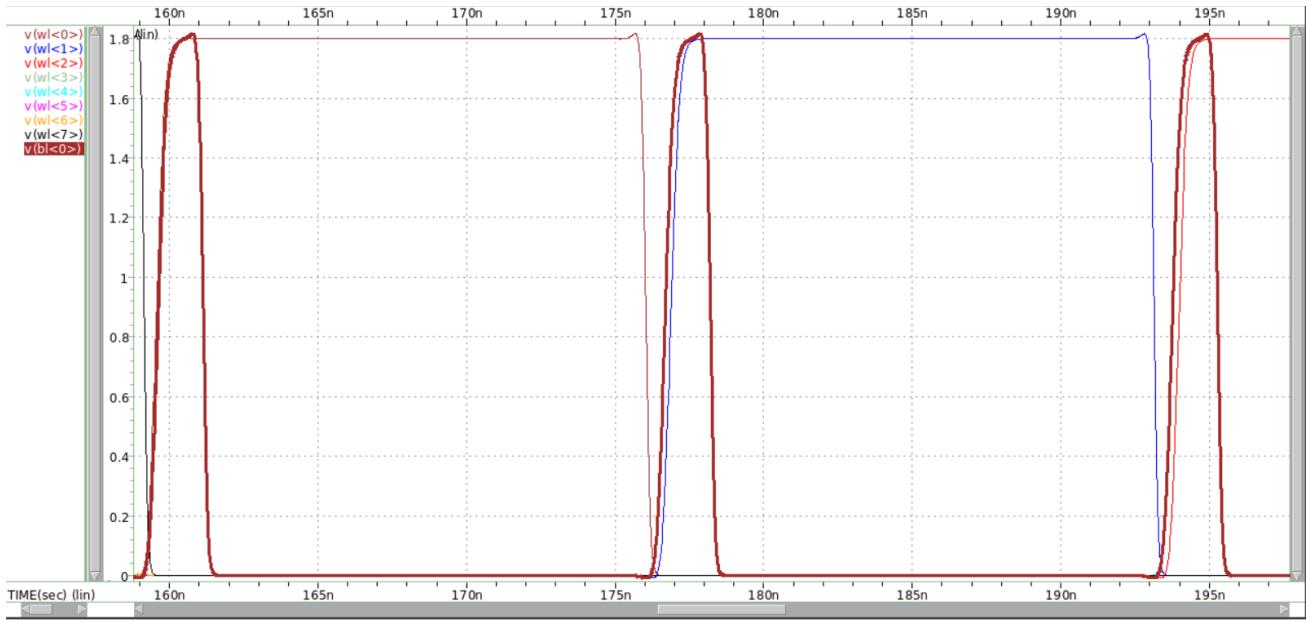
可以看出 reset、enable、decoder、frequency divider 皆有正常運作，此外沒有 glitch 的產生(h)、可以看出 bit line 的改變由 f_2 決定(i)。



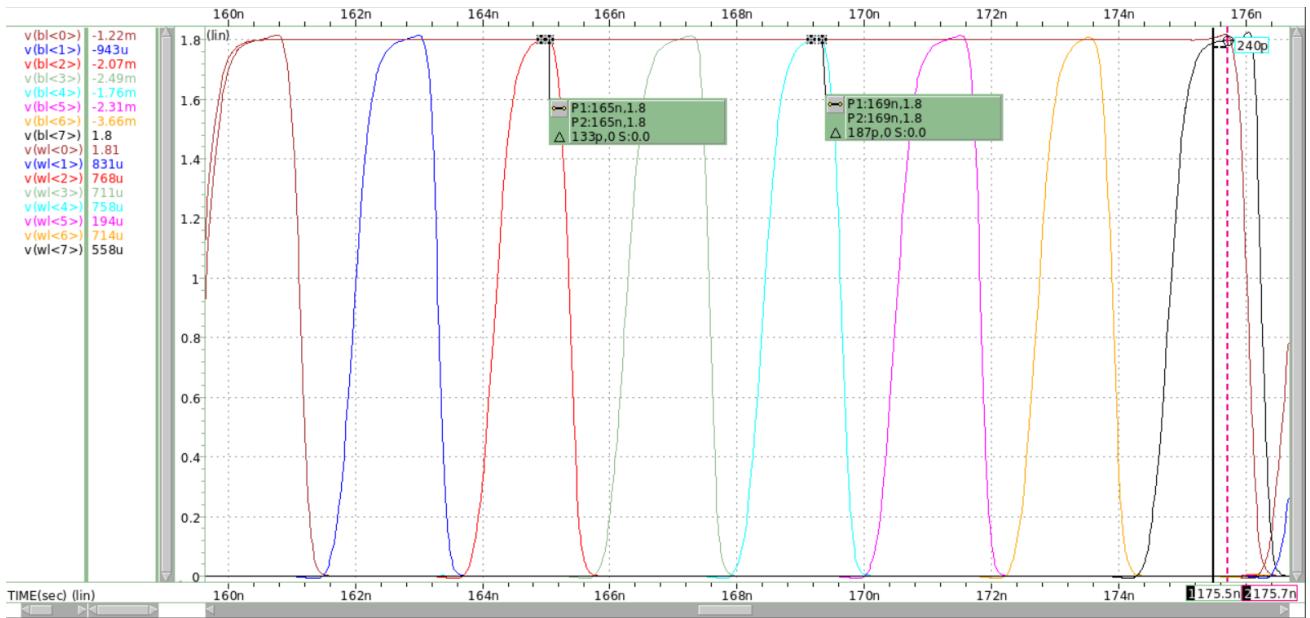
- (a) When $BL< n+1 >$ turn on , $BL< n >$ must turn off, intersection of two lines " $V \leq 0.9(V)$ "



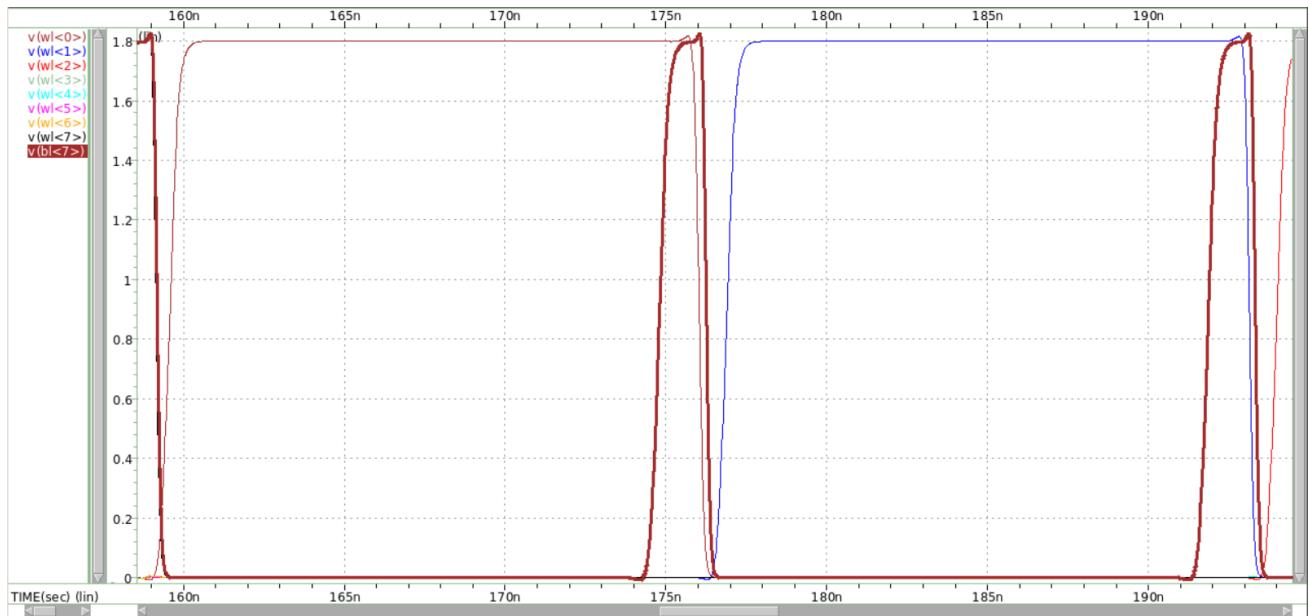
- (b) When $WL<0>$ turn on , $BL<0>$ must be ready. (1.8(V))
- (d) When $BL<0>$ turn on , $WL<n-1>$ must turn off, intersection of two lines " $V \leq 0.4(V)$ "



- (c) Each BL must overlap with WL for more than 0.1(nsec) *note : Both are 1.8 (V)



- (e) We recommend that $WL<7>$ be designed to shut down earlier than $BL<7>$
- (f) When $WL<7>$ turn on , $BL<7>$ must turn off , intersection of two lines " $V \leq 0.4(V)$ "



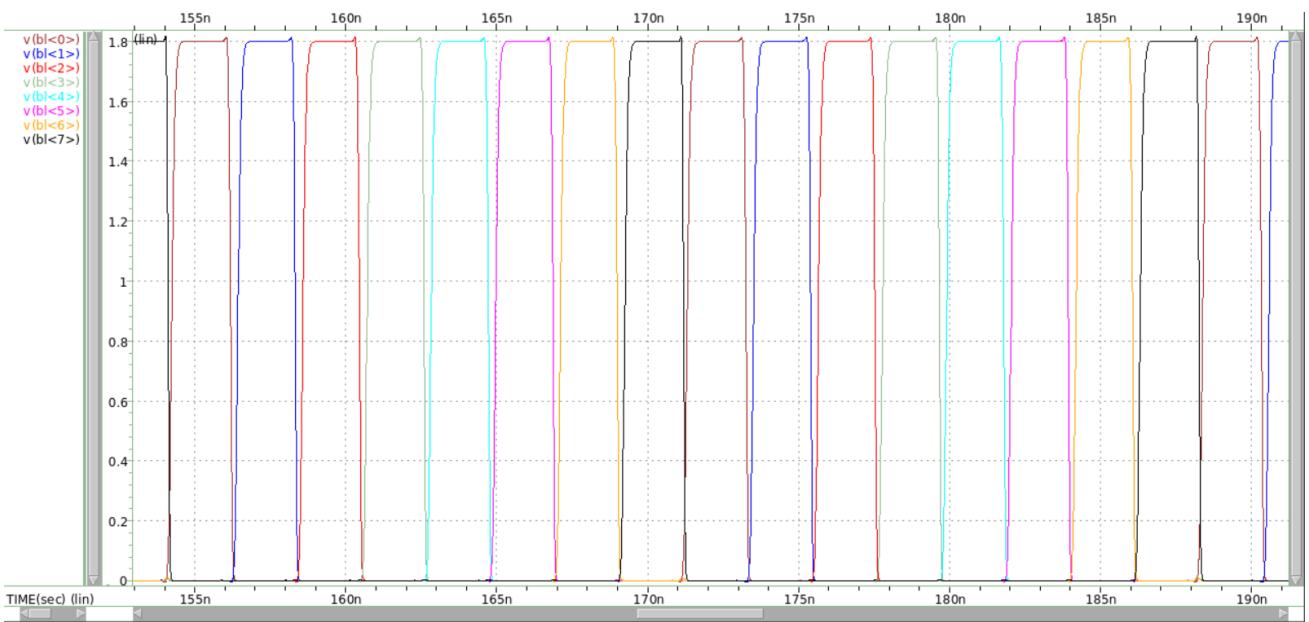
(5) FF corner

結果總圖：

可以看出 reset、enable、decoder、frequency divider 皆有正常運作，此外沒有 glitch 的產生(h)、可以看出 bit line 的改變由 f_2 決定(i)。

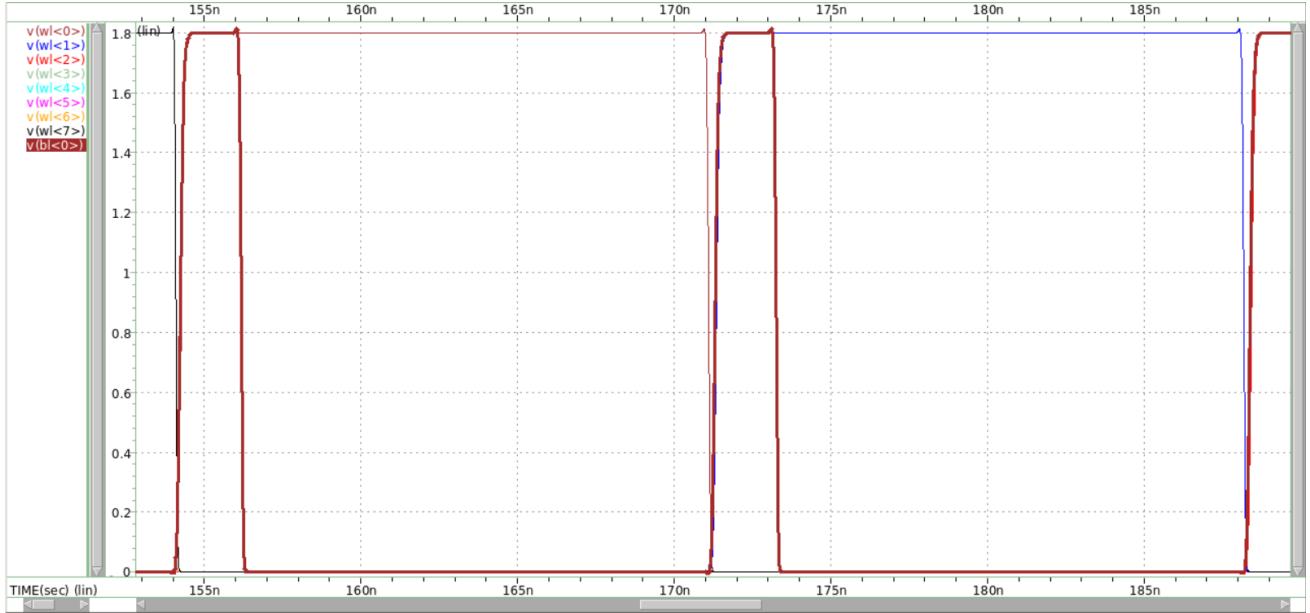


- (a) When $BL< n+1 >$ turn on , $BL< n >$ must turn off, intersection of two lines " $V \leq 0.9(V)$ "

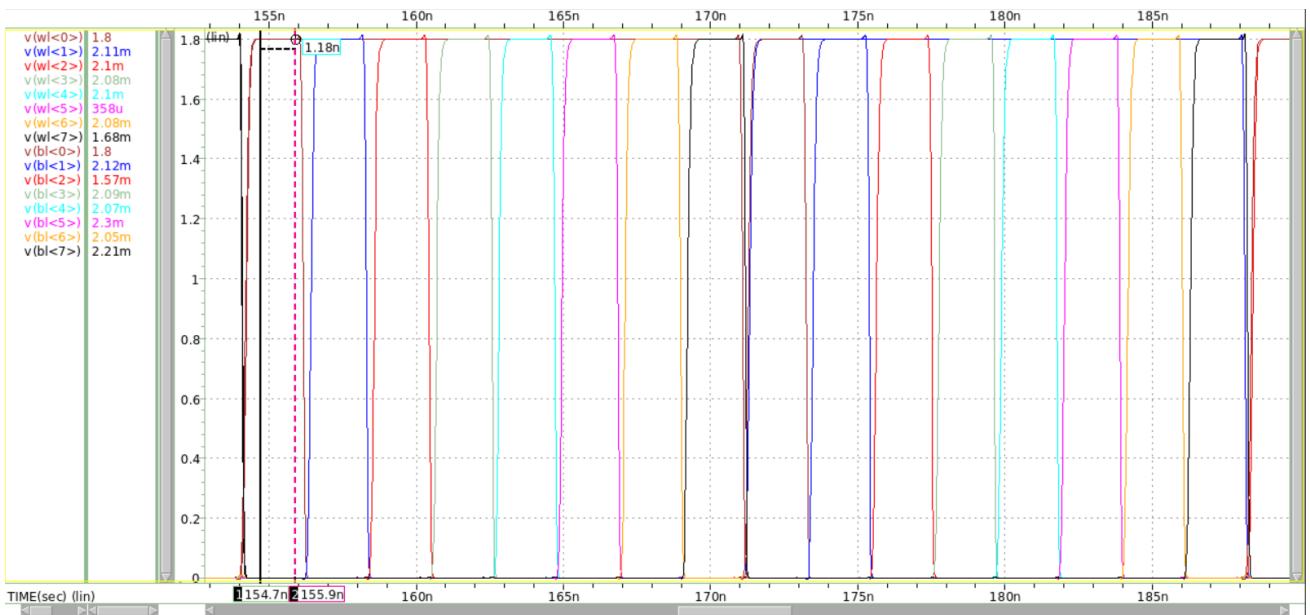


- (b) When $WL<0>$ turn on , $BL<0>$ must be ready. (1.8(V))

- (d) When $BL<0>$ turn on , $WL <n-1>$ must turn off, intersection of two lines " $V \leq 0.4(V)$ "

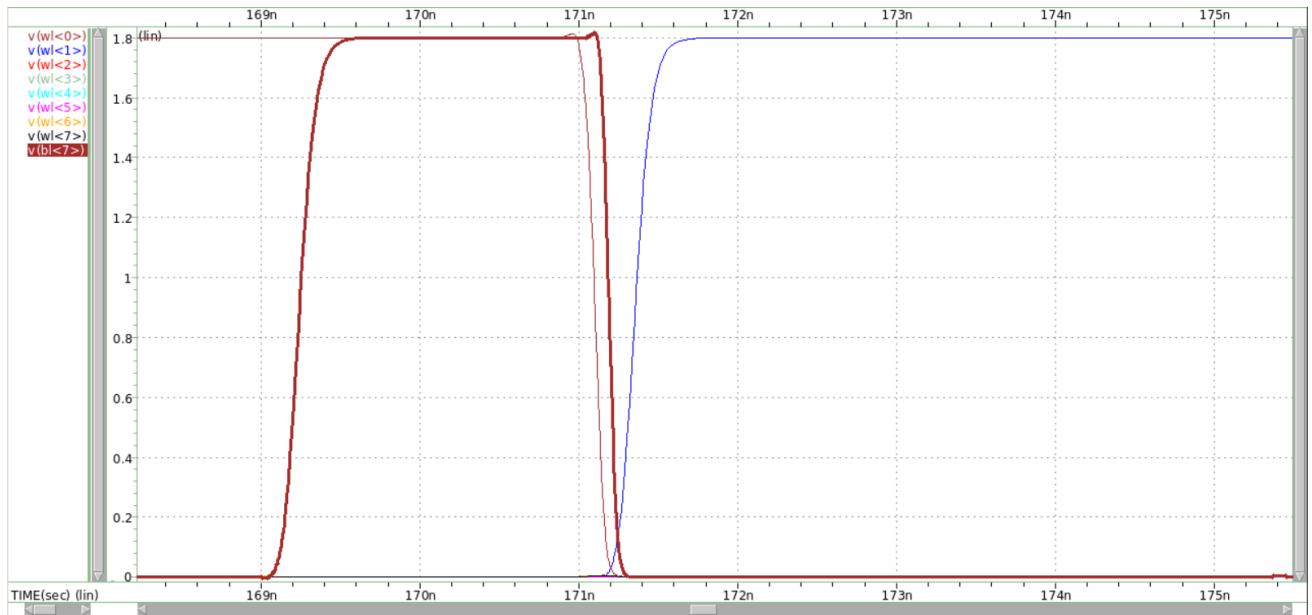
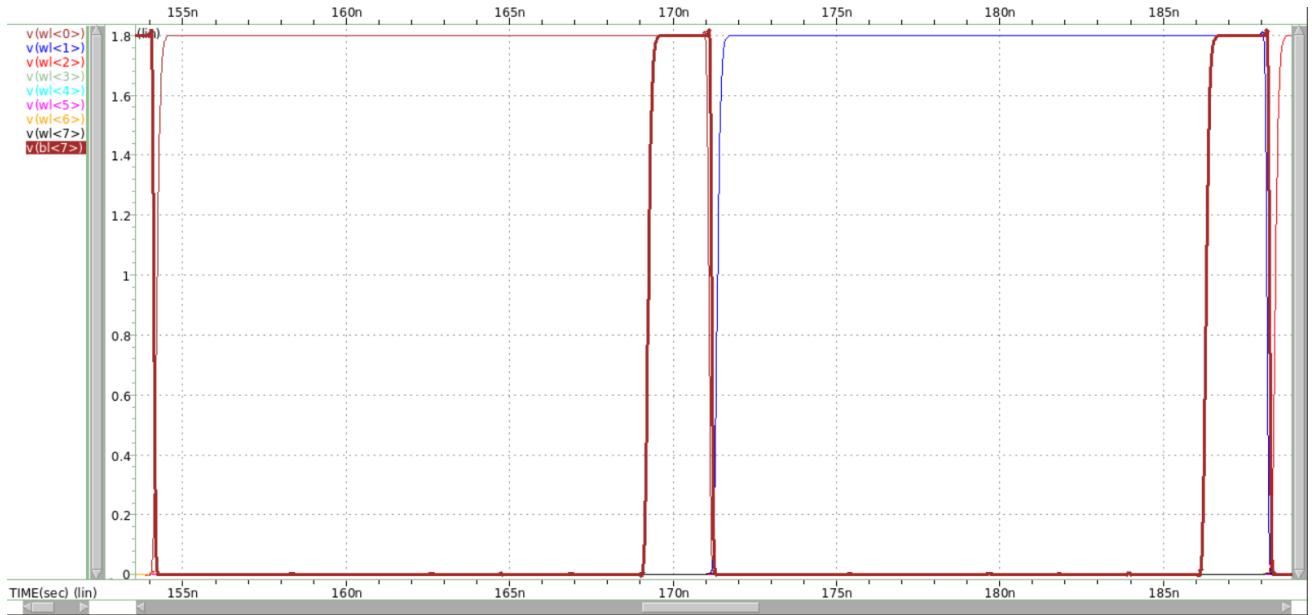


- (c) Each BL must overlap with WL for more than $0.1(nsec)$ *note : Both are 1.8 (V)



- (e) We recommend that $WL<7>$ be designed to shut down earlier than $BL<7>$

- (f) When $WL<7>$ turn on, $BL<7>$ must turn off, intersection of two lines " $V \leq 0.4(V)$ "



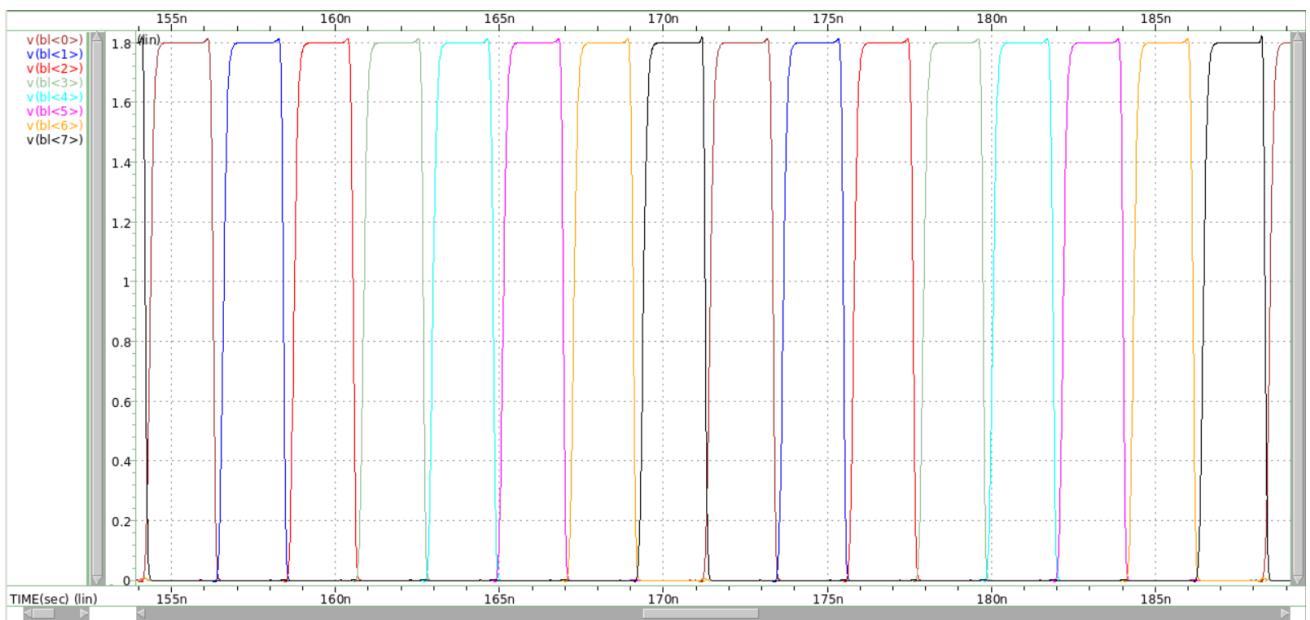
(6) FS corner

結果總圖：

可以看出 reset、enable、decoder、frequency divider 皆有正常運作，此外沒有 glitch 的產生(h)、可以看出 bit line 的改變由 f_2 決定(i)。



- (a) When $BL< n+1 >$ turn on , $BL< n >$ must turn off , intersection of two lines " $V \leq 0.9(V)$ "

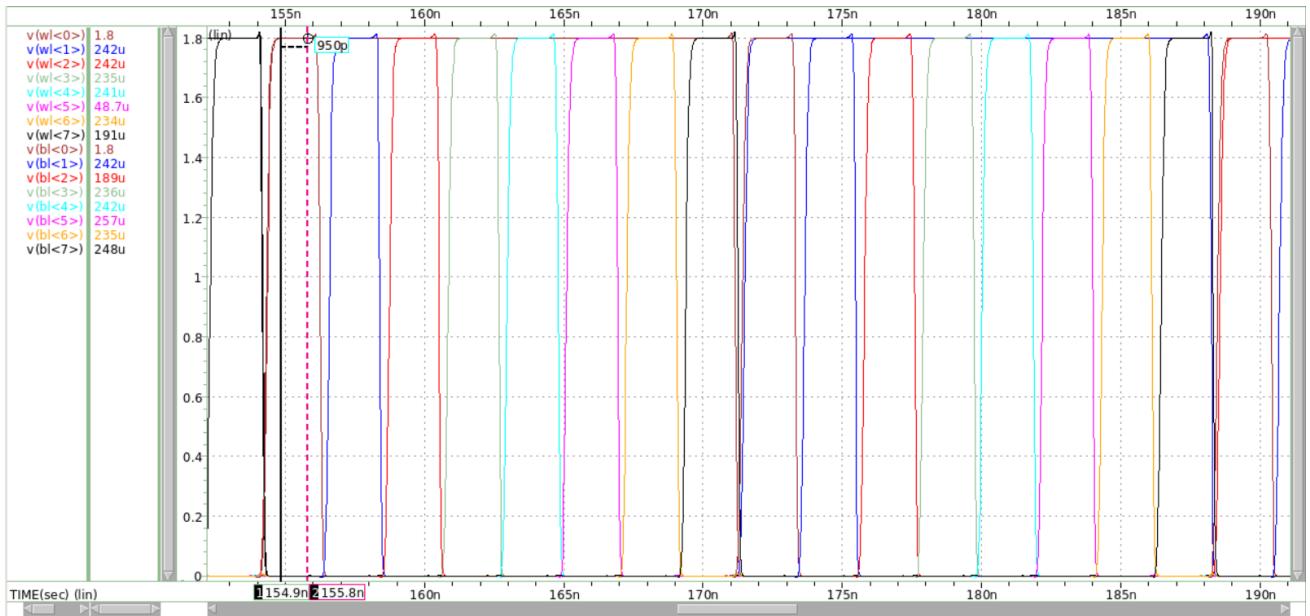


- (b) When $WL<0>$ turn on , $BL<0>$ must be ready. (1.8(V))

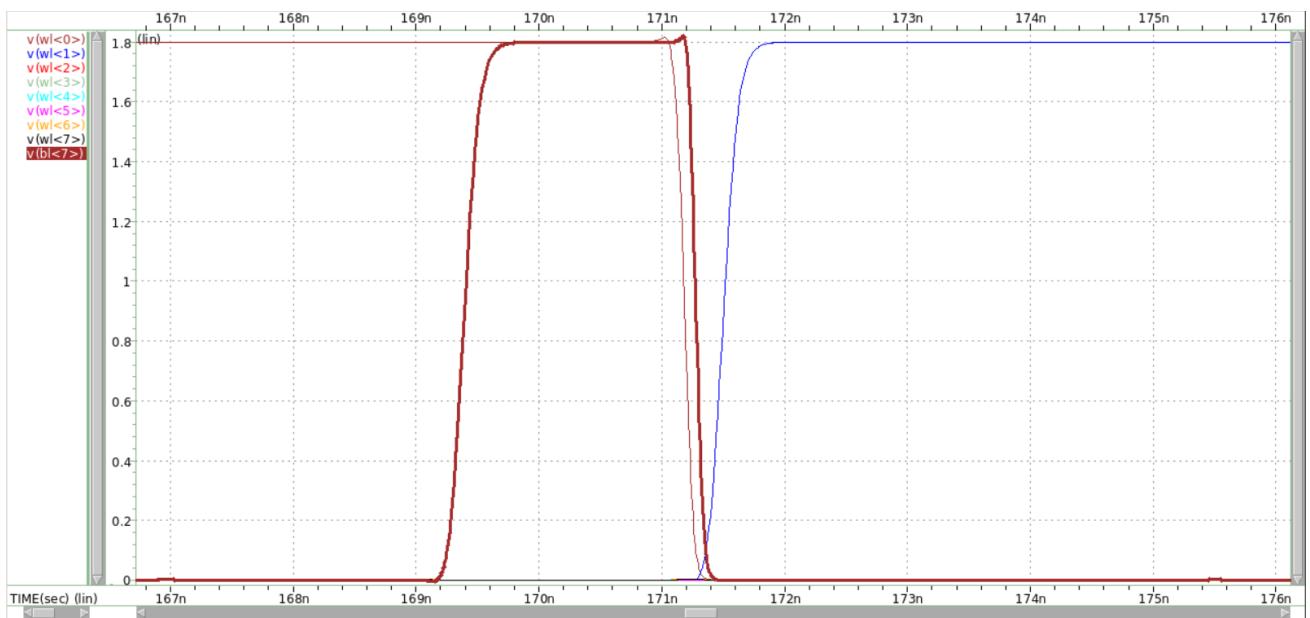
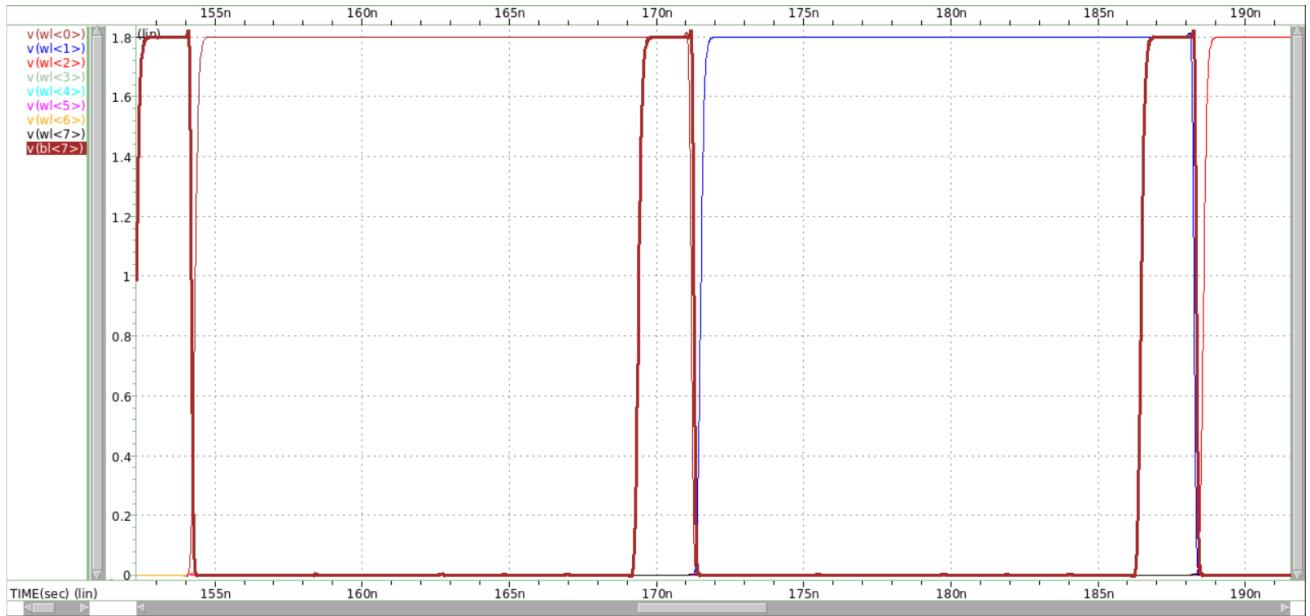
- (d) When $BL<0>$ turn on , $WL <n-1>$ must turn off, intersection of two lines " $V \leq 0.4(V)$ "



- (c) Each BL must overlap with WL for more than 0.1(nsec) *note : Both are 1.8 (V)



- (e) We recommend that $WL< n >$ be designed to shut down earlier than $BL< 7 >$
- (f) When $WL< n+1 >$ turn on , $BL< 7 >$ must turn off , intersection of two lines " $V \leq 0.4(V)$ "



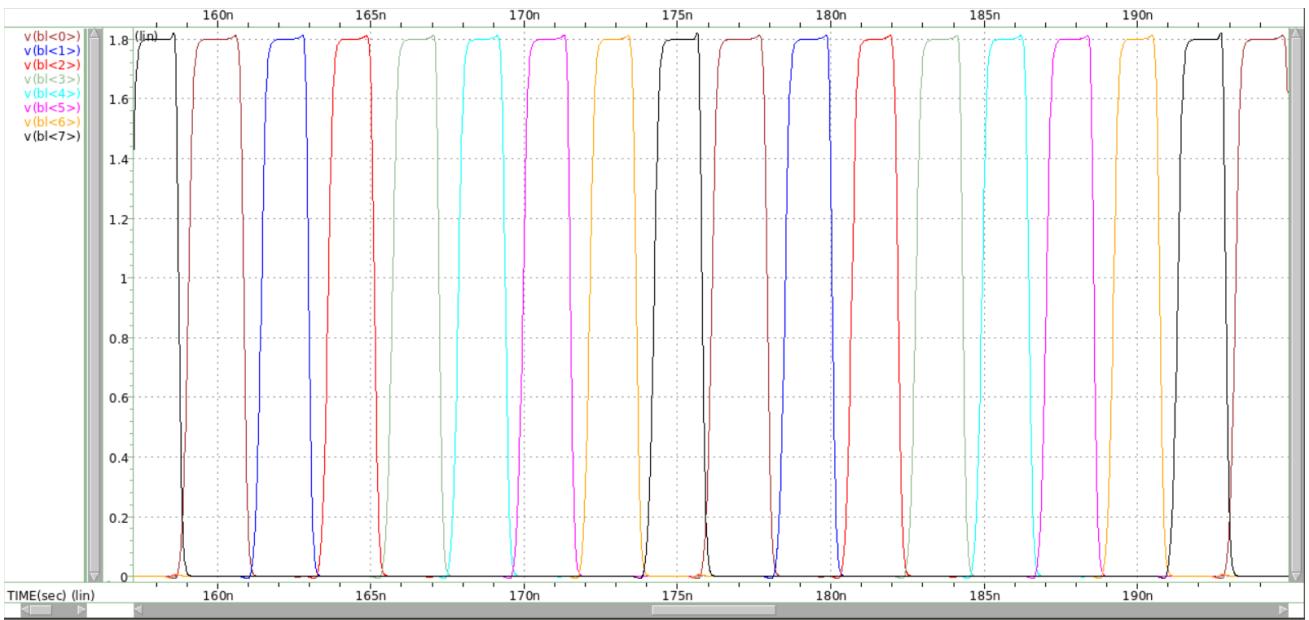
(7) SF corner

結果總圖：

可以看出 reset、enable、decoder、frequency divider 皆有正常運作，此外沒有 glitch 的產生(h)、可以看出 bit line 的改變由 f_2 決定(i)。

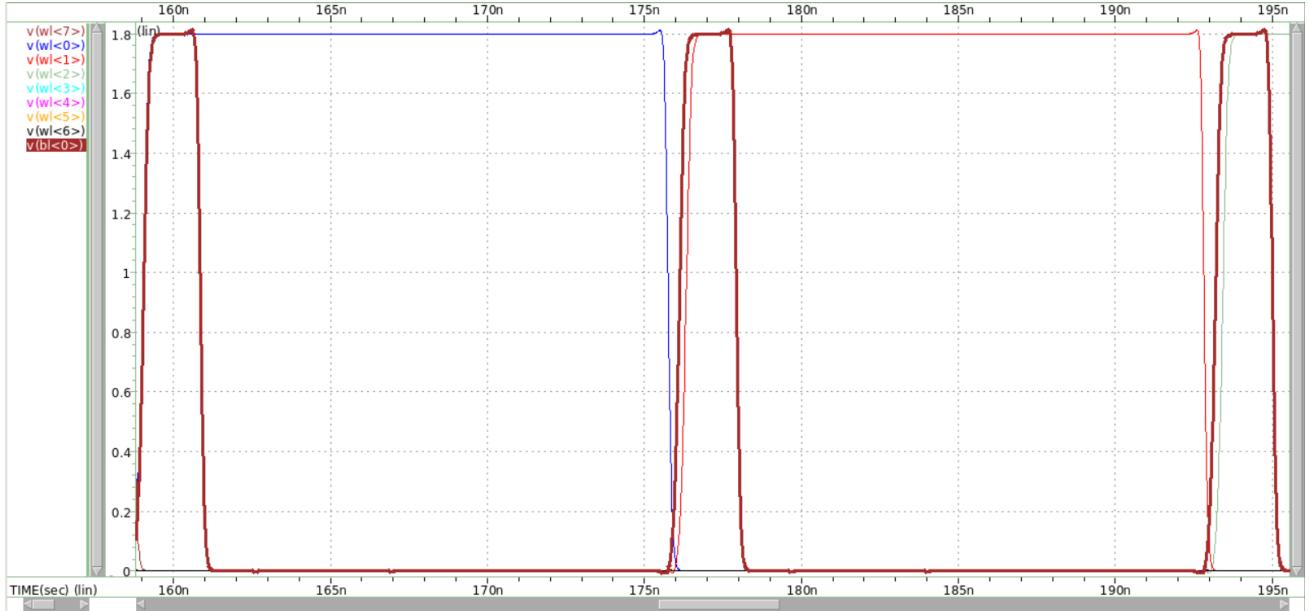


- (a) When $BL< n+1 >$ turn on , $BL< n >$ must turn off , intersection of two lines " $V \leq 0.9(V)$ "

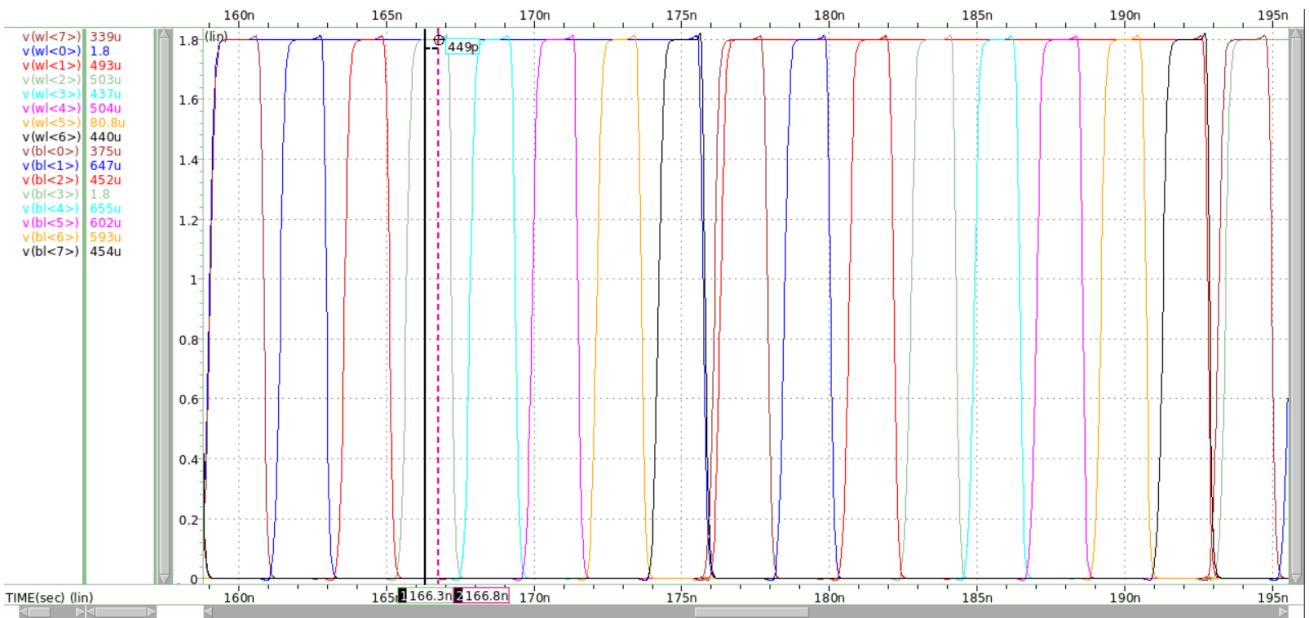


- (b) When $WL<7>$ turn on , $BL<0>$ must be ready. (1.8(V))

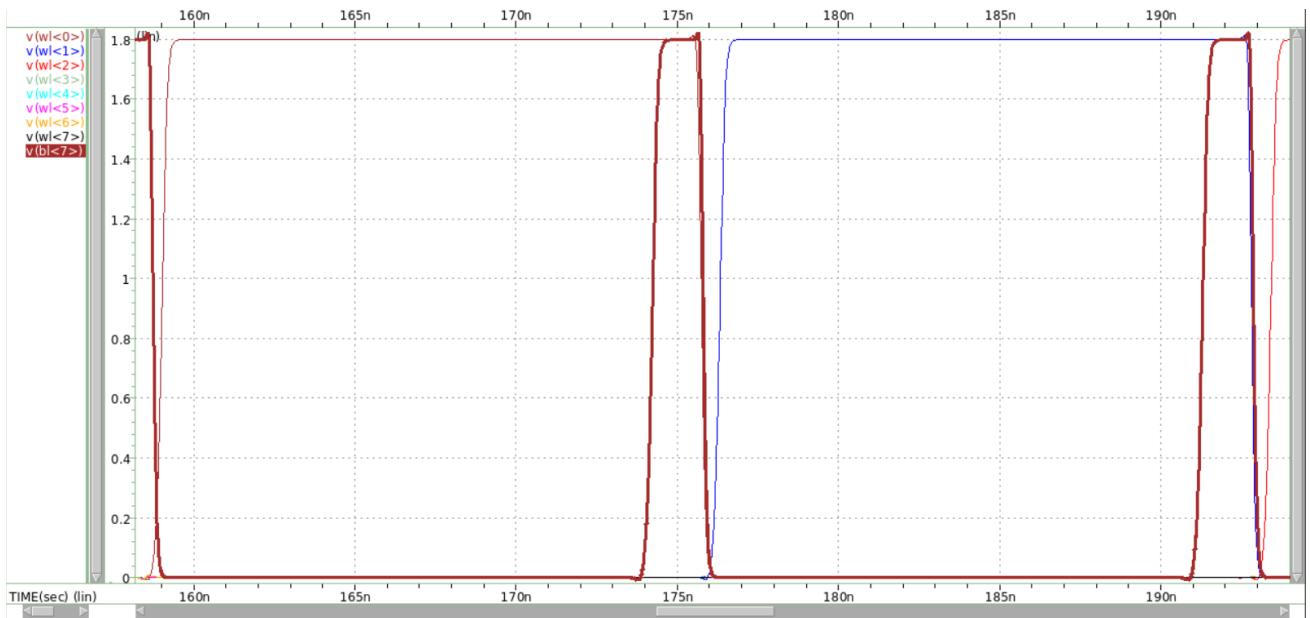
- (d) When $BL<0>$ turn on , $WL<7>$ must turn off, intersection of two lines " $V \leq 0.4(V)$ "



- (c) Each BL must overlap with WL for more than 0.1(nsec) *note : Both are 1.8 (V)

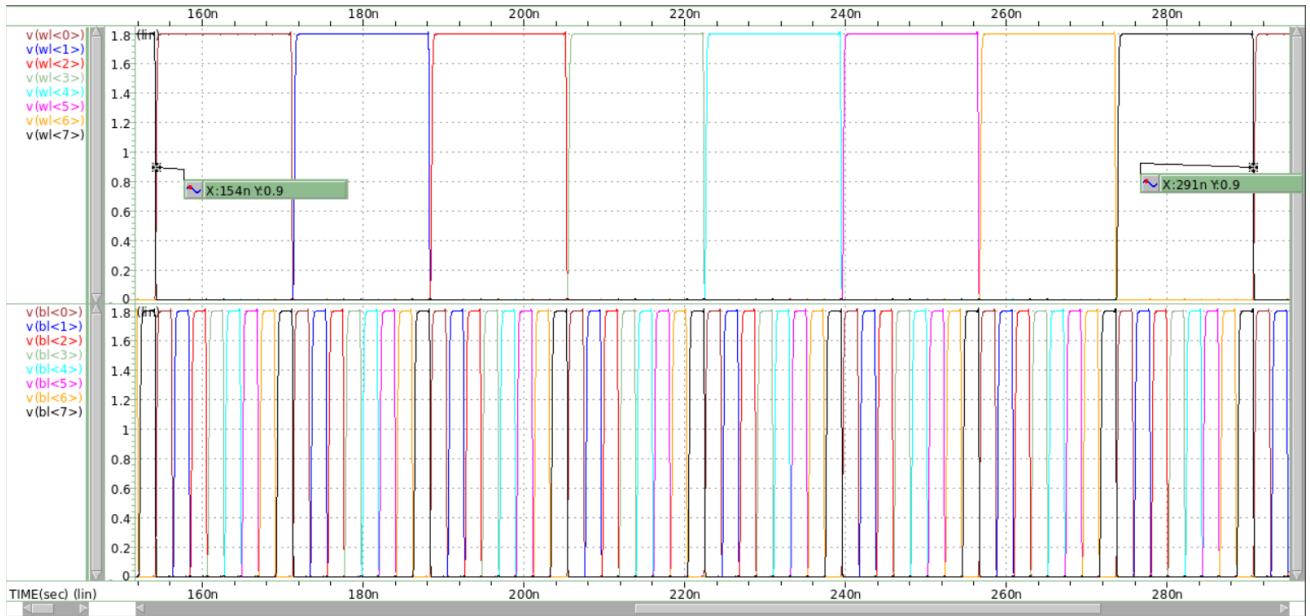


- (e) We recommend that $WL<7>$ be designed to shut down earlier than $BL<7>$
- (f) When $WL<7>$ turn on, $BL<7>$ must turn off, intersection of two lines " $V \leq 0.4(V)$ "



(8) Power consumption

首先先觀察 TT corner 中一次完整的轉換(wl<0> ~ wl<7>)的起終點：



因此使用.meas 指令去對 power 取平均測量：

```
*****Analysis*****
.op
.tran "(1/1000)*2.2*pw" 350ns
.meas tran average_power AVG power from 154.3n to 290.7n
```

結果如下：

```
***** transient analysis tnom= 25.000 temp= 25.000 *****
average_power= 446.2023u from= 154.3000n to= 290.7000n
```

(9) FoM 計算

將頻率 470 MHz、面積 836.7 μm^2 、TT corner 平均 Power 0.45 mW 代入 FoM 的

$$\text{Performance} = \frac{\text{Max operation frequency}^2}{\text{area}^2} \div \text{power}$$

計算公式
中，可以

得到最終的 FoM 結果為 0.70。

3. Pre-sim 與 Post-sim 的結果差異

整體而言，我們有發現 Post-sim 與 Pre-sim 在結果的 pattern 上幾乎沒有差別，也就是說並沒有因為畫 Layout 導致某個部件的寄生電阻電容特別大，而導致功能上出現某個情況下無法運作的情況，換句話說，也就是每個部件的寄生電阻電容在跑過 PEX 後是均勻的增加的，這使得最終的結果功能性上沒有因為 Layout 而影響。

但當然的，Post-sim 後由於寄生電阻電容的增加，整體的運作頻率有因此下降約 15%，從原本的 550 MHz 變為 470 MHz，且此頻率的下降是無可避免的。

最後想要由 FoM 的角度再次出發，雖說 Post-sim 後頻率會因此下降，但由 power 公式可以知道 power 會正比於頻率，因此 power 也會下降約 15%左右，對於 FoM 的影響方面，頻率與 power 一來一回的結果就是 FoM 只下降 15%左右，這也再次應證了面積才是最為影響 FoM 的因素，驗證了我們此次設計 final_project 的核心目標是正確的。