

VLSI Circuits Design HW#5

Course	Name	Student ID
[3] 2-4	陳旭祺	E24099059

● 1A - 6T SRAM

Please design a 6T-SRAM(as shown in Fig.1) with the clock signal(CLK) and control signals (WE and RE). WE is write enable mode, and RE is read enable mode. You should use **pre-charge circuit**, **read circuit** and **write circuit** to enable 1-bit READ and WRITE function work correctly.

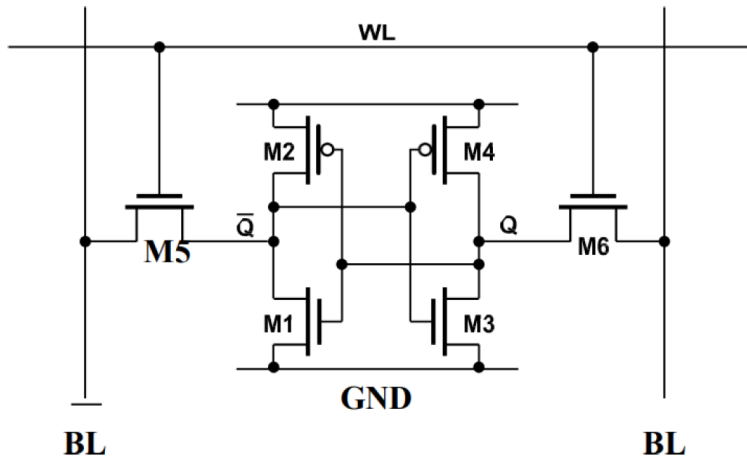


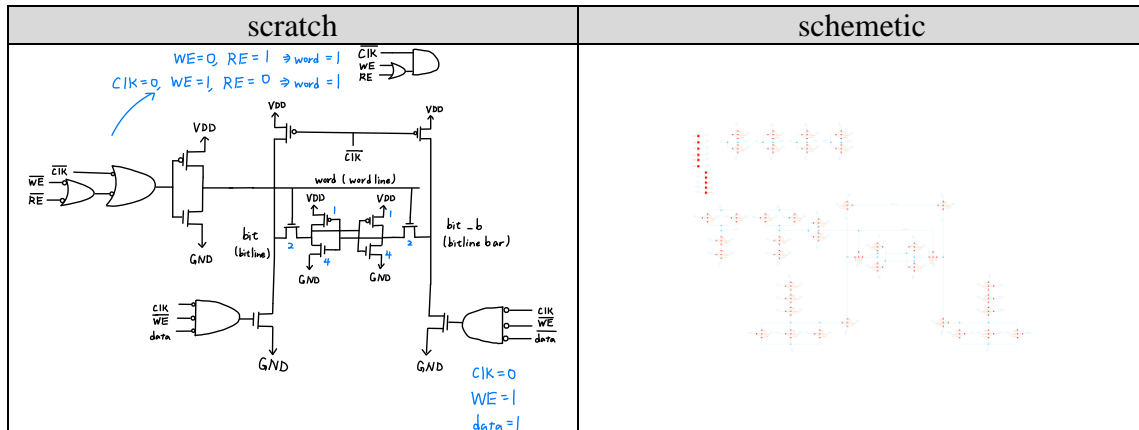
Fig.1 6T SRAM cell

You should simulate the following four patterns to check your design can work correctly:

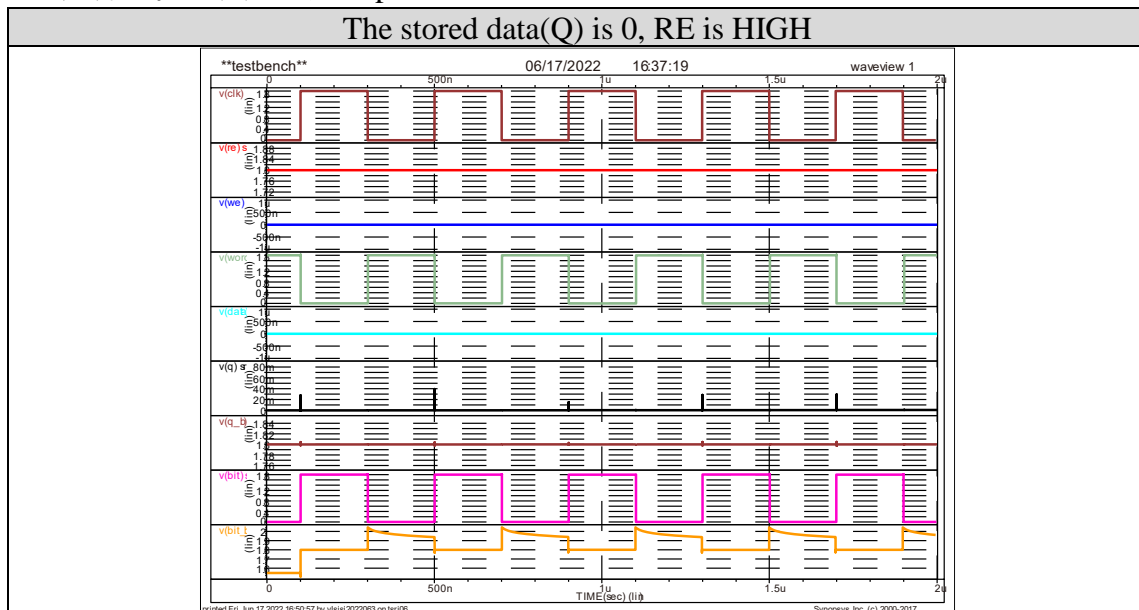
1. The stored data(Q) is 0, RE is HIGH.
2. The stored data(Q) is 1, RE is HIGH.
3. The stored data(Q) is 0, WE is HIGH, write data is 1.
4. The stored data(Q) is 1, WE is HIGH, write data is 0.

Please paste the simulation result in your report with the following signals: CLK, WE, RE, WL, Vwrite_data, BL, BLB, Q, and QB.

除了6T SRAM cell之外還有周邊電路 - bitline conditioning、write driver，控制 wordline訊號的邏輯閘需要考慮。其中write driver電路我採用由組合訊號控制的單一電晶體，雖然比起用2顆電晶體實現的方法所花費電晶體數量較多，但是電路阻抗較小，因此較能驅動PMOS transistor，才能寫入0。最後SRAM cell的電晶體尺寸設計，我使用講義U46-17的範例比例。

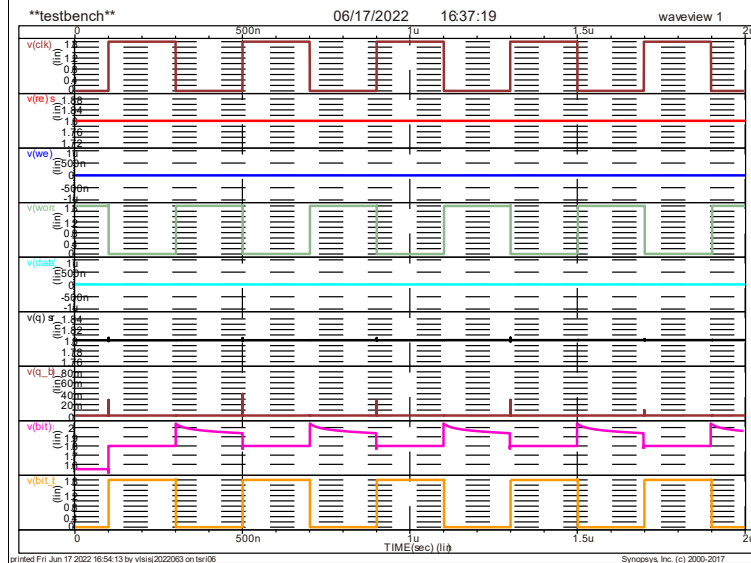


分別測量題目要求以下4個pattern



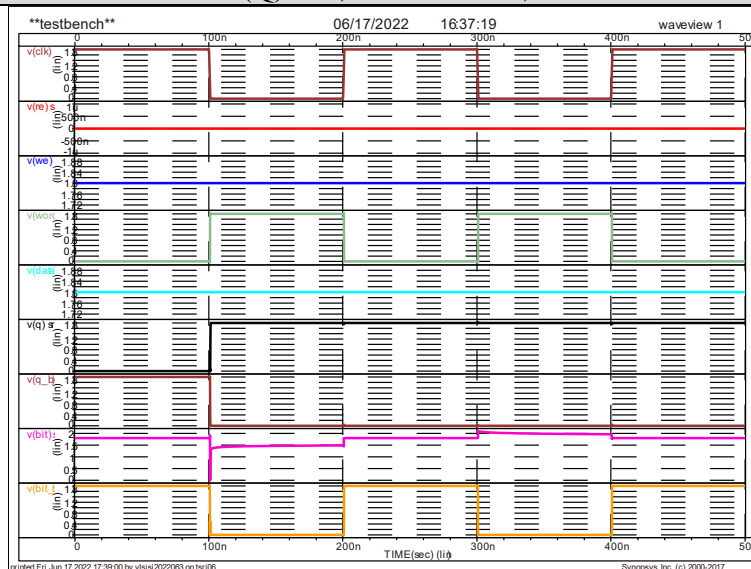
read operation中 $CLK = 1$ 時bitline conditioning電路的PMOS打開，會先把bit和bit_b都precharge到 V_{DD} ，接下來 $CLK = 0$ 時，PMOS關閉bit和bit_b變為floating，再來wordline從0變為1，把N2、N4打開，由於cross-coupled inverter的 $Q = 0$ ，會把原本precharge到 V_{DD} 的bit往下拉至0，代表把data讀出來，因此內部電晶體要比外部讀取電晶體還strong， $N1 > N2, N3 > N4$ ，此外bit_b由於floating，會受到wordline從0變為1的耦合電容，因此電壓衝高於 V_{DD} 。

The stored data(Q) is 1, RE is HIGH.

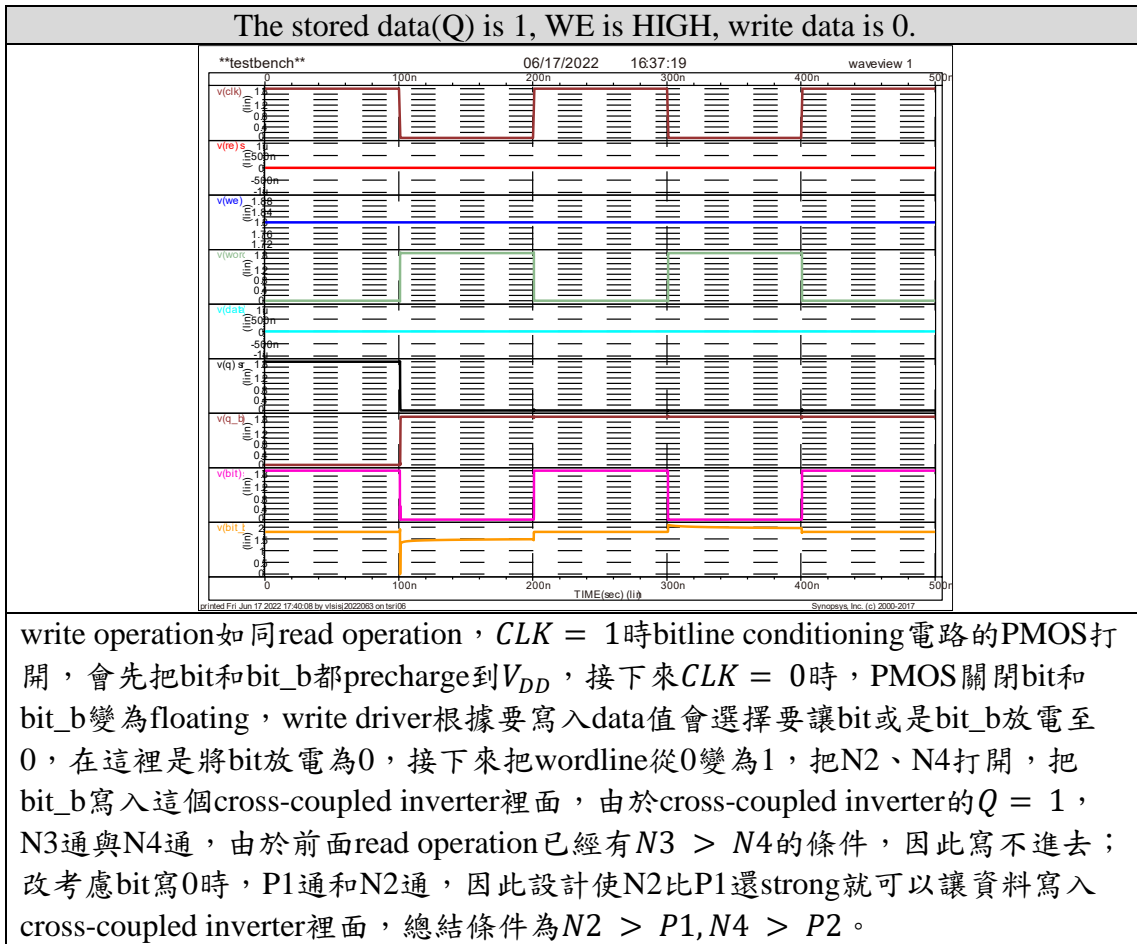


read operation 中 $CLK = 1$ 時 bitline conditioning 電路的 PMOS 打開，會先把 bit 和 bit_b 都 precharge 到 V_{DD} ，接下來 $CLK = 0$ 時，PMOS 關閉 bit 和 bit_b 變為 floating，再來 wordline 從 0 變為 1，把 N2、N4 打開，由於 cross-coupled inverter 的 $Q_b = 1$ ，會把原本 precharge 到 V_{DD} 的 bit_b 往下拉至 0，代表把 data 讀出來，因此內部電晶體要比外部讀取電晶體還 strong， $N1 > N2, N3 > N4$ ，此外 bit 由於 floating，會受到 wordline 從 0 變為 1 的耦合電容，因此電壓衝高於 V_{DD} 。

The stored data(Q) is 0, WE is HIGH, write data is 1.



write operation 如同 read operation， $CLK = 1$ 時 bitline conditioning 電路的 PMOS 打開，會先把 bit 和 bit_b 都 precharge 到 V_{DD} ，接下來 $CLK = 0$ 時，PMOS 關閉 bit 和 bit_b 變為 floating，write driver 根據要寫入 data 值會選擇要讓 bit 或是 bit_b 放電至 0，在這裡是將 bit_b 放電為 0，接下來把 wordline 從 0 變為 1，把 N2、N4 打開，把 bit 寫入這個 cross-coupled inverter 裡面，由於 cross-coupled inverter 的 $Q_b = 1$ ，N1 通與 N2 通，由於前面 read operation 已經有 $N1 > N2$ 的條件，因此寫不進去；改考慮 bit_b 寫 0 時，P2 通和 N4 通，因此設計使 N4 比 P2 還 strong 就可以讓資料寫入 cross-coupled inverter 裡面，總結條件為 $N2 > P1, N4 > P2$ 。

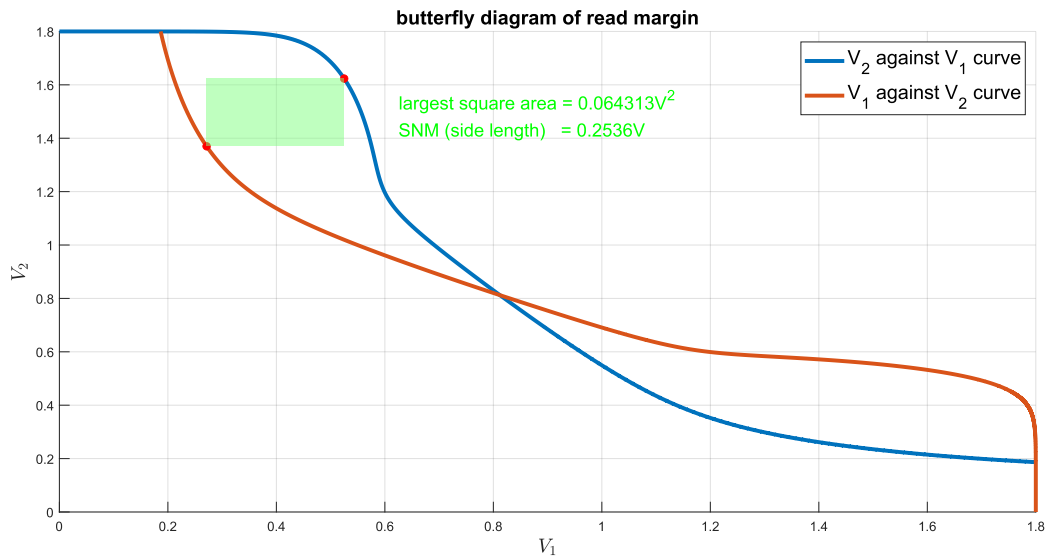


● 1B - read margin of 6T SRAM

DC分析掃描輸入的值並下關鍵字.print將要觀測的值輸出至*.lis文字檔，手動修改文字檔內原本的單位 m, μ, n 為 $e-3, e-6, e-9$ ，同VLSI作業三，使用matlab指令readmatrix，將文字檔轉為矩陣形式，對matlab來說只要能轉換成矩陣格式之後數據就好處理了，將原圖形x, y對調並與原圖做疊圖，得下圖。

畫出來的butterfly diagram後，找出曲線之間所圍出來最大正方形面積的邊長，即為static noise margin(SNM)，意義為施予多少noise下會使cross-cupled inverter失去原來stable的狀態，如下圖一read margin經程式計算即為0.2536V和下圖二write margin經程式計算即為0.5281V，需要注意的是由於下圖x與y軸尺度不一致，所以看起來不像正方形，但實際為正方形。

VLSI lab5



```

plot(v1, v2,'LineWidth', 3);
plot(v2, v1,'LineWidth', 3);

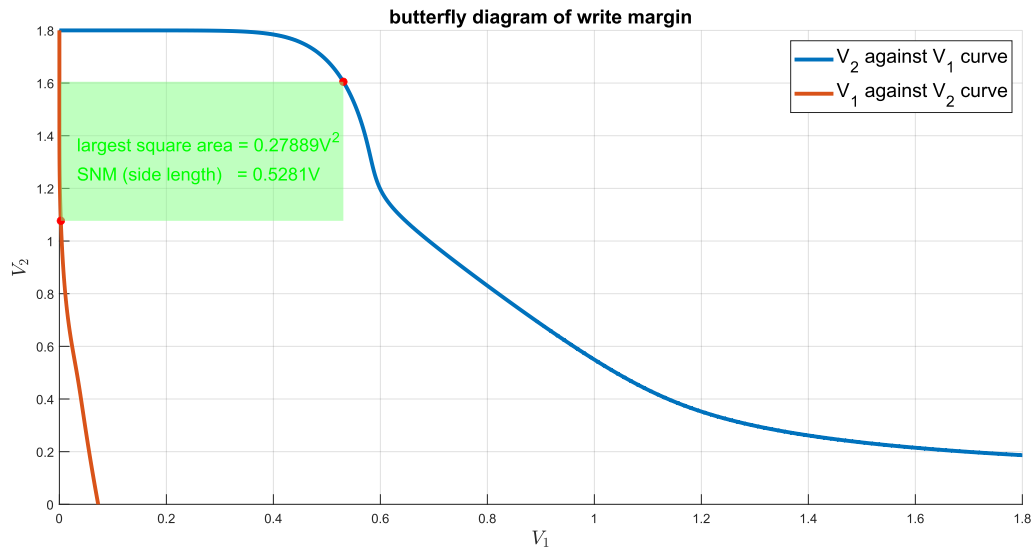
%% find SNM
max_area = 0;
index_point = zeros(1, 2);
n = length(v1);
while v1(n) > 0.8
    k = length(v1) - n + 1;
    while v2(k) > 0.8
        cal_area = abs(v1(k) - v2(n))*abs((v2(k) - v1(n))); %
area = abs(x1 - x2)abs(y1 - y2)
        if (max_area < cal_area) && (v1(k) > v2(n)) && (v2(k) >
v1(n)) && (abs(v1(k) - v2(n)) == abs((v2(k) - v1(n))))
            max_area = cal_area;
            index_point(1) = n;
            index_point(2) = k;
        end
        k = k + 1;
    end
    n = n - 1;
end

```

令藍線上的點為 $(v1(k), v2(k))$ ，橘線上的點為 $(v2(n), v1(n))$ ，第一層while loop從橘線左上角掃動至右下角的點，一直到橘線低於 $0.8V$ ，也就是在交點附近的位置，第二層while loop在固定橘線上的點的前提下，藍線跟橘線一樣從左上角掃動至右下角的點，接下來計算兩點之間的計算面積，只有在掃動到更大值、藍線上的點在橘線上的點右上角、圍出來是正方形，這三個條件成立下才會更新max_area，並記錄該點的index存入一開始preallocated好的index_point矩陣，方便後續程式抓取線上點的位置。

● 1C - write margin of 6T SRAM

同1B小題做法，繪圖如下



```

plot(x1, y1,'LineWidth', 3);
plot(y2, x2,'LineWidth', 3);

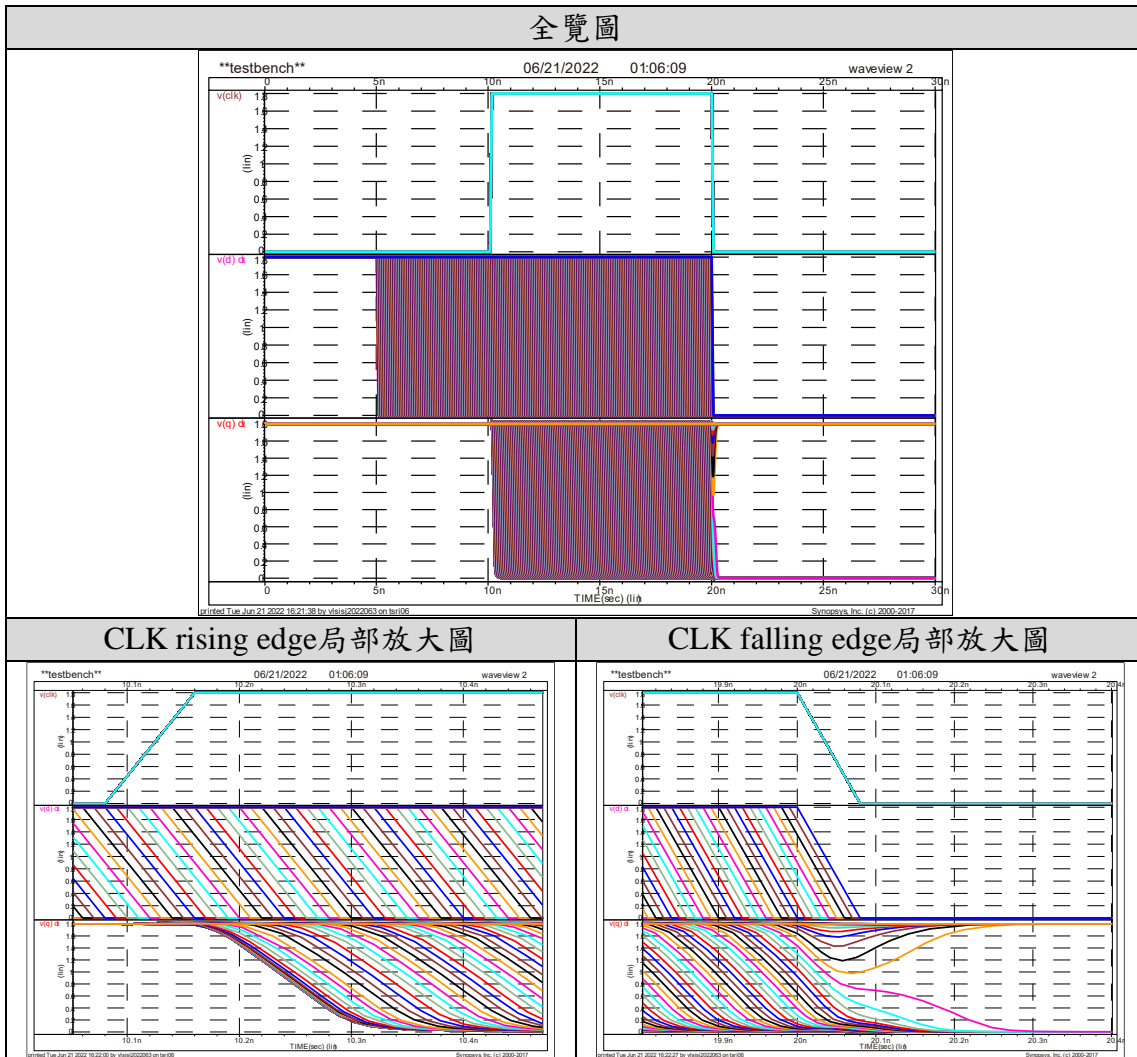
%% find SNM
max_area = 0;
index_point = zeros(1, 2);
for n = 1:length(y2)
    for k = 1:length(y1)
        cal_area = abs(x1(k) - y2(n))*abs(y1(k) - x2(n)); %
area = abs(x1 - x2)abs(y1 - y2)
        if (max_area < cal_area) && (x1(k) > y2(k)) &&
(abs((x1(k) - y2(n))) == abs(y1(k) - x2(n)))
            max_area = cal_area;
            index_point(1) = n;
            index_point(2) = k;
        end
    end
end
end

```

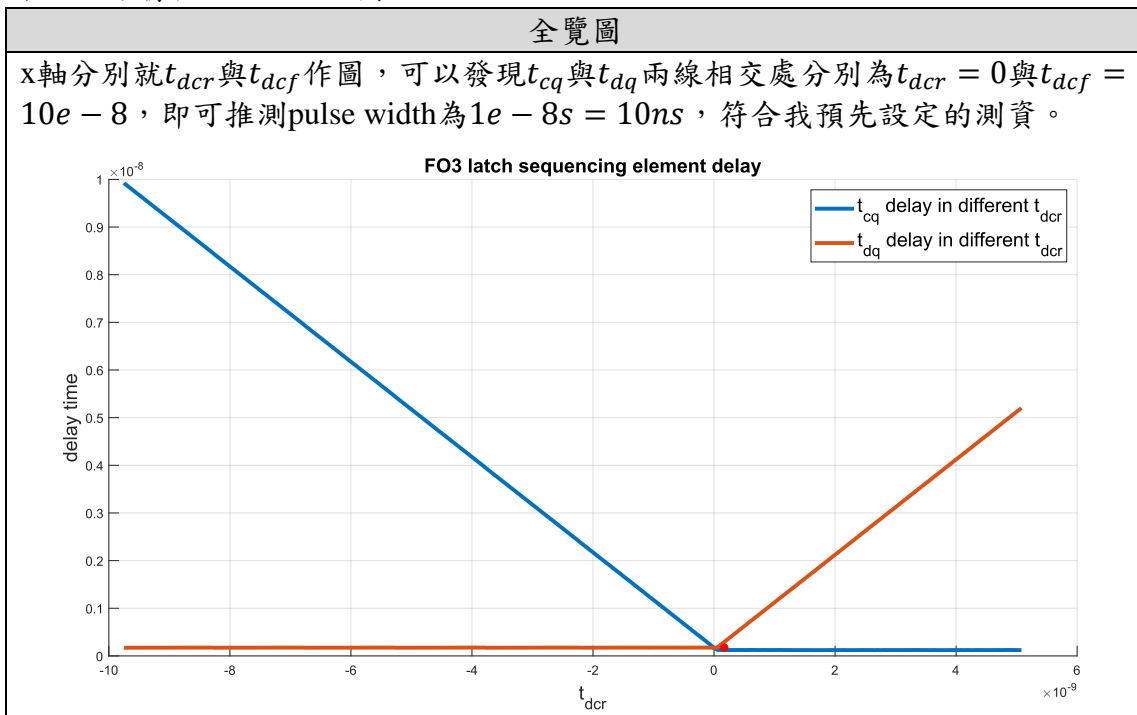
令藍線上的點為 $(x1(k), y1(k))$ ，橘線上的點為 $(y2(n), x2(n))$ ，第一層for loop從橘線右下角掃動至左上角的點，第二層for loop在固定橘線上的點的前提下，從藍線左上角掃動至右下角的點，接下來計算兩點之間的計算面積，只有在掃動到更大值、藍線上的點在橘線上的點右邊、圍出來是正方形，這三個條件成立下才會更新max_area，並記錄該點的index存入一開始preallocated好的index_point矩陣，方便後續程式抓取線上點的位置。

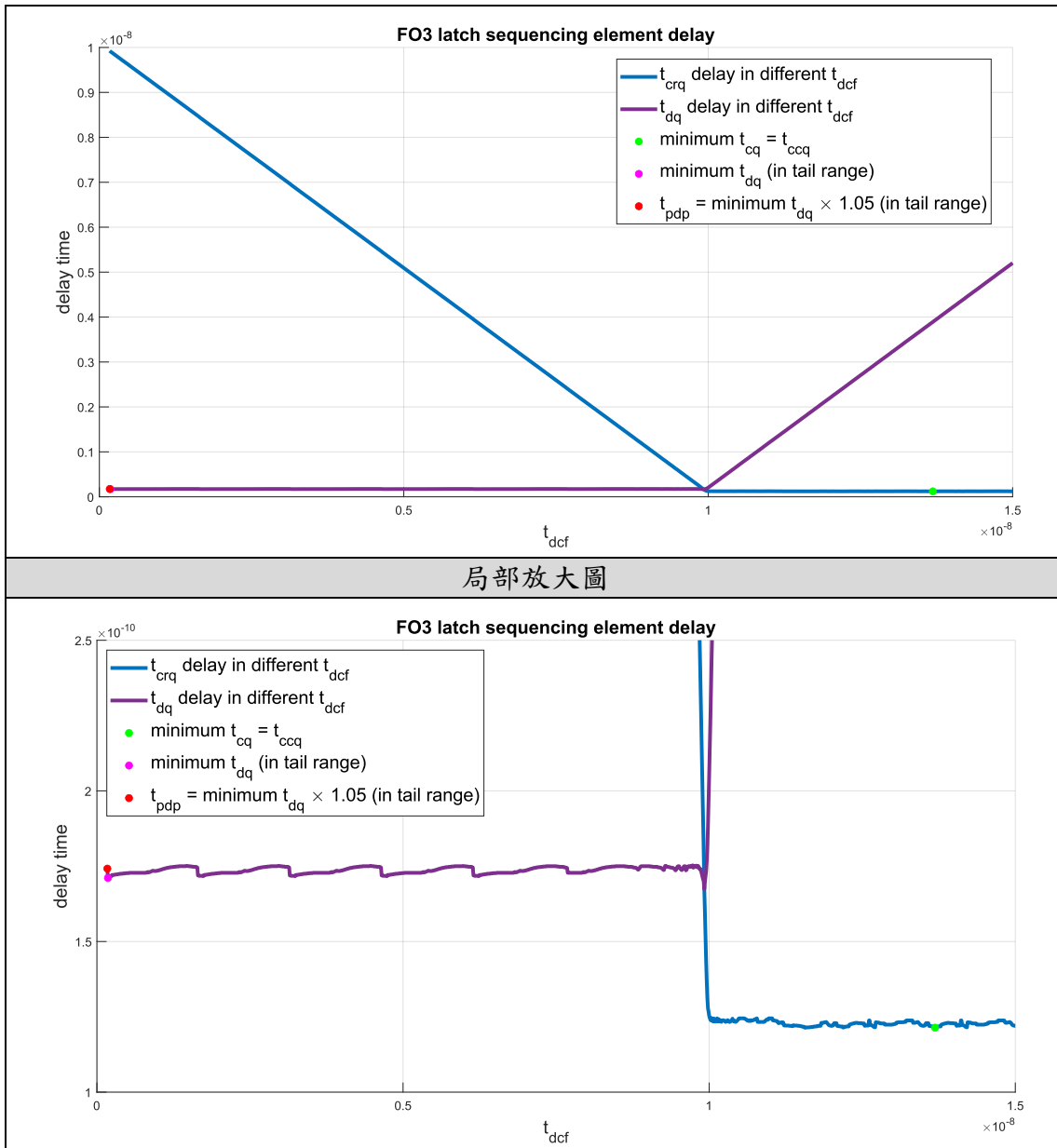
● 2B - sequencing element delay of latch

輸入D使用PWL訊號，改變下降的時間點去sweep掃值，data需要在CLK的falling edge來臨前的一段時間以上setup好，輸出Q才可以成功轉態，這個時間即為latch的setup time，否則一旦違反setup time，輸入D就無法傳到輸出Q。



將*mt0數據匯入matlab做圖





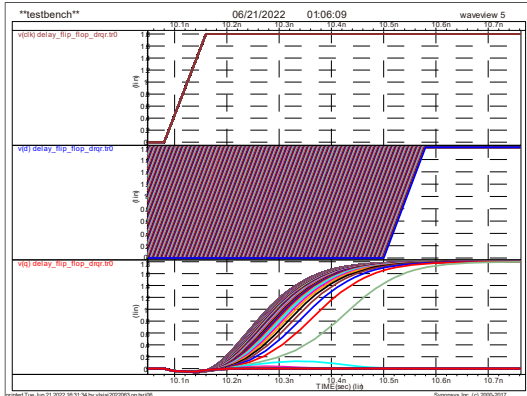
● 2A - find setup time of latch

t_{setup}	t_{ccq}	t_{pdq}
0.170ns	0.121ns	0.179ns

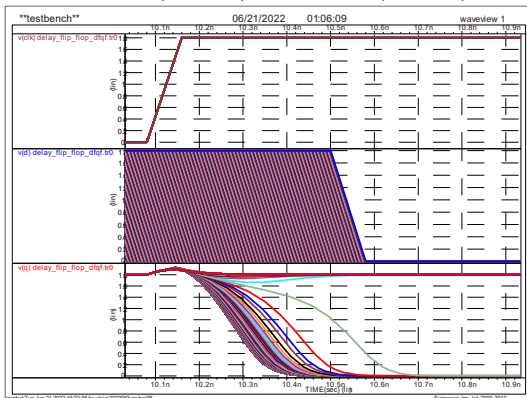
t_{cq} 最小值即為 t_{ccq} ， t_{dq} 最小值再乘上 1.05 即為 t_{pdp} ， t_{pdp} 該點對應的 t_{dcf} 值大小就是 t_{setup} time。如上圖所示，推測由於 latch 沒有明顯的 transient state，因此隨著 t_{dcf} 下降， t_{dq} 沒有 delay 往上升的趨勢， t_{dq} 在 t_{dcf} 小於 $2.4e - 10$ 數值會直接測不到 failed，對應到 2B 小題 CLK falling edge 局部放大圖中輸出 D 第一條沒成功轉態的橘色線，由於大於 0.9V，不會 trigger 到此波型，因此沒有測量到此數據而 failed，綜合以上原因， t_{setup} time 我取最後一筆沒有 failed 的 t_{dcf} 值。

● 2D - sequencing element delay of flip flop

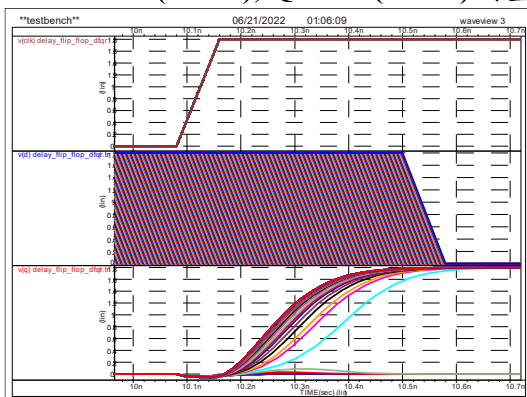
sequencing element delay 作圖需要測量 4 條線，在不同 t_{dc} 下的 t_{cq} 延遲時間差，依序為

A. D rises (0 → 1), Q rises (0 → 1) 測量 t_{setup1} 

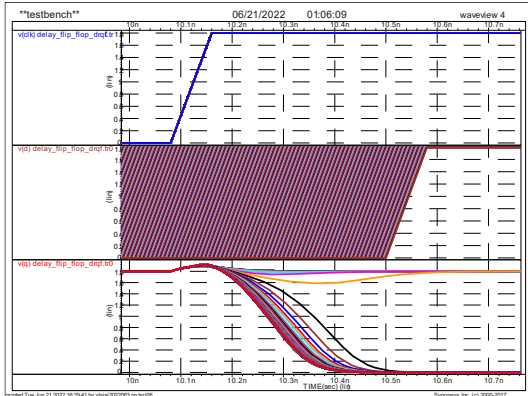
給定Q初始值低電位，在正常情況下CLK positive trigger會抓取輸入D的高電位使得輸出Q電位從設定好的初始值低電位變為高電位，但是隨著 t_{dc} 減少，D的rising edge往右移，當輸入D小於 $t_{dc} = t_{\text{setup1}}$ 之後才rise，輸出Q會變成較難轉態，如上圖綠線， t_{dc} 繼續減少一直到輸出無法成功轉態而維持在原本的低電位。

B. D falls (1 → 0), Q falls (1 → 0) 測量 t_{setup0} 

給定Q初始值高電位，在正常情況下CLK positive trigger會抓取輸入D的低電位使得輸出Q電位從設定好的初始值高電位變為低電位，但是隨著 t_{dc} 減少，D的falling edge往右移，當輸入D小於 $t_{dc} = t_{\text{setup0}}$ 之後才fall，輸出Q會變成較難轉態，如上圖綠線， t_{dc} 繼續減少一直到輸出無法成功轉態而維持在原本的高電位。

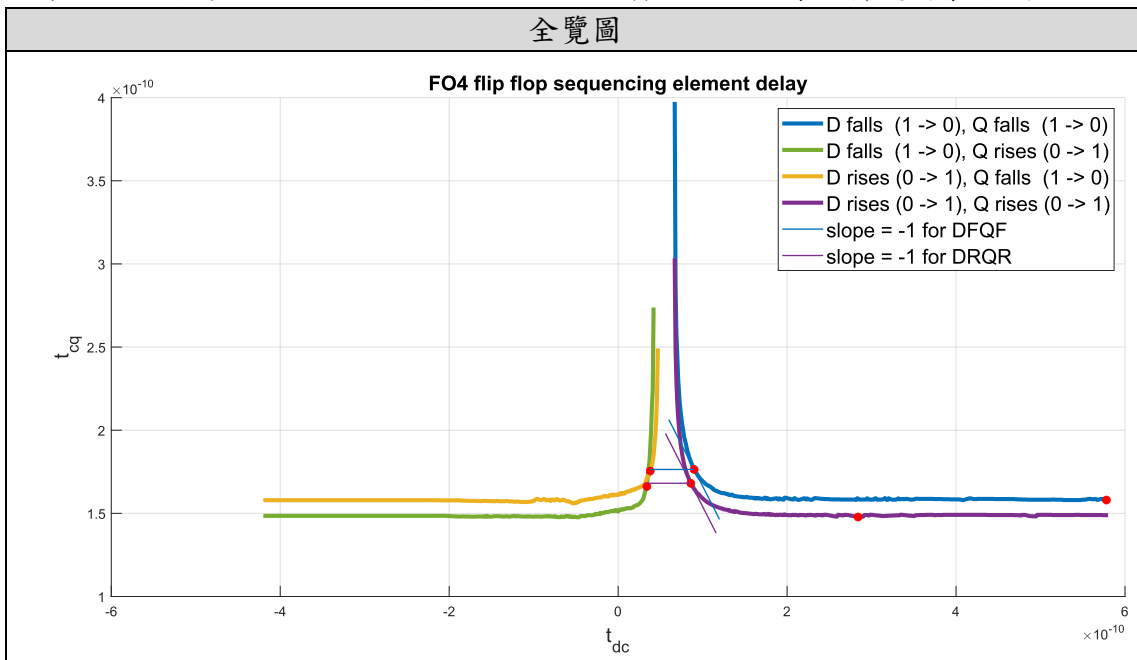
C. D falls (1 → 0), Q rises (0 → 1) 測量 $-t_{\text{hold1}}$ 

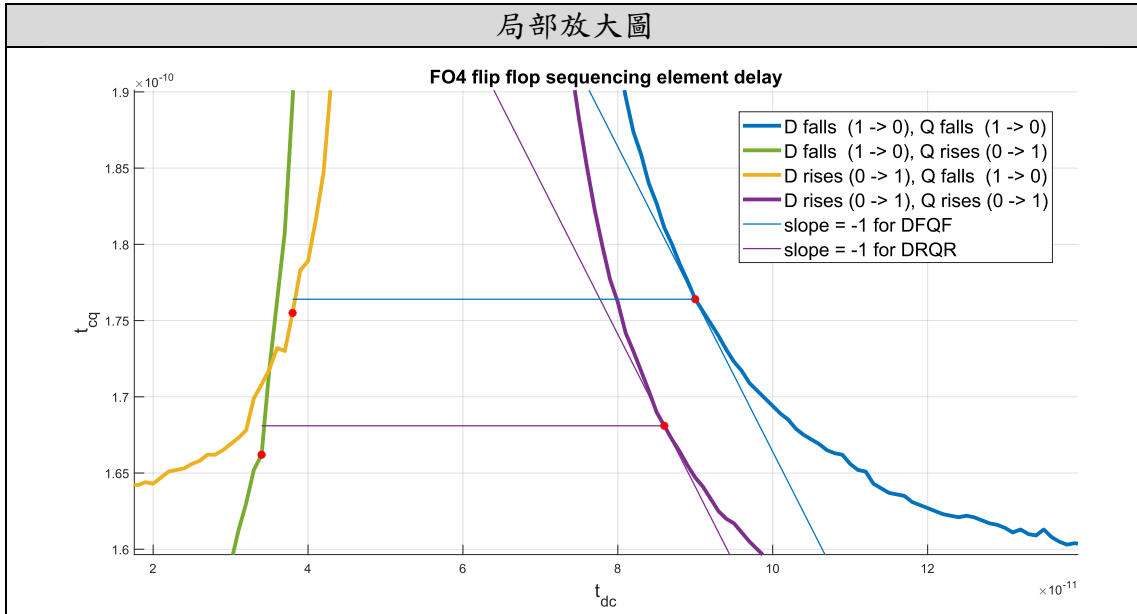
給定Q初始值低電位，在正常情況下CLK positive trigger會抓取輸入D的高電位使得輸出Q電位從設定好的初始值低電位變為高電位，但是隨著 t_{dc} 增加，D的falling edge往左移，當輸入D大於 $t_{dc} = -t_{\text{hold1}}$ 之後才fall(hold time是負號代表發生在CLK rising edge之前)，輸出Q會變成較難轉態，如上圖淡藍色線， t_{dc} 繼續加大一直到輸出無法成功轉態而維持在原本的低電位。

D.D rises (0 -> 1), Q falls (1 -> 0)測量 $-t_{hold0}$ 

給定Q初始值高電位，在正常情況下CLK positive trigger會抓取輸入D的低電位使得輸出Q電位從設定好的初始值高電位變為低電位，但是隨著 t_{dc} 增加，D的rising edge往左移，當輸入D大於 $t_{dc} = -t_{hold0}$ 之後才rise(hold time是負號代表發生在CLK rising edge之前)，輸出Q會變成較難轉態，如上圖黑線， t_{dc} 繼續加大一直到輸出無法成功轉態而維持在原本的高電位。

測完數據後是分析結果，我原本跟前幾次作業一樣匯出文字數據到*mt0檔，使用matlab指令readmatrix讀檔，但是不知道什麼原因無法讀檔，更改方法為打開waveview軟體，選擇指定數值作圖，再點軟體介面左上角export，將數據輸出為純文字檔，接下來使用matlab指令readmatrix讀檔，做數據分析後繪製下圖





● 2C - find setup time and hold time of flip flop

t_{setup0}	t_{pcq0}	t_{ccq0}	$-t_{hold0}$
0.0900ns	0.1764ns	0.1580ns	0.0380ns
t_{setup1}	t_{pcq1}	t_{ccq1}	$-t_{hold1}$
0.086ns	0.1681ns	0.1478ns	0.0340ns
t_{ar}	t_{af}		
0.048ns	0.056ns		

在 t_{DC} 較大處定義 t_{ccq} ，代表意義為在clock來之前flip flop內部節點就已經well settled，當clock一來，訊號馬上傳過去， t_{cq} 會是最小值，因此這邊我取DFQF和DRQR兩條線的最小值，如全覽圖2個紅點標記。

DFQF和DRQR兩條線 t_{cq} 在斜率為-1時代表在 t_{DQ} 的極(小)值，並於此點定義 t_{setup}, t_{pcq} ，我使用數據點1與數據點3、數據點2與數據點4等間隔為1的兩點取斜率，並與-1取距離，距離最近者代表斜率最接近-1，取此兩數據點的中間值即為要找的 $(t_{dc}, t_{cq}) = (t_{setup}, t_{pcq})$ 。

```

slope_dfqf = zeros(length(tcq_dfqf)-2, 1);
for n = 1:length(tcq_dfqf)-2
    slope_dfqf(n) = (tcq_dfqf(n) - tcq_dfqf(n + 2))/2e-12;
end
slope_dfqf_abs = abs(slope_dfqf(:) - (-1));
[setup_dfqf_error, setup_dfqf_index] = min(slope_dfqf_abs);
setup_dfqf_index = setup_dfqf_index + 1;

```

接下來在維持與setup time相同的sequencing overhead $t_{cq} = t_{pcq}$ 前提之下，代表y軸固定，水平線往左延伸至DRQF和DRQR兩條線，由於數據點是離散的，所以要找離 $t_{cq} = t_{pcq}$ 水平線最近的點，該點即為 $t_{dc} = -t_{hold}$ 。

有了以上這些數據就可以計算rising input的aperture width $t_{ar} = t_{setup1} + t_{hold0}$ 和falling input的aperture width $t_{af} = t_{setup0} + t_{hold1}$ ，代表意義為在clock edge前後會有一段時間data不可改變，如果改變可能會造成metastable或是無窮時間的delay。

● Conclusion

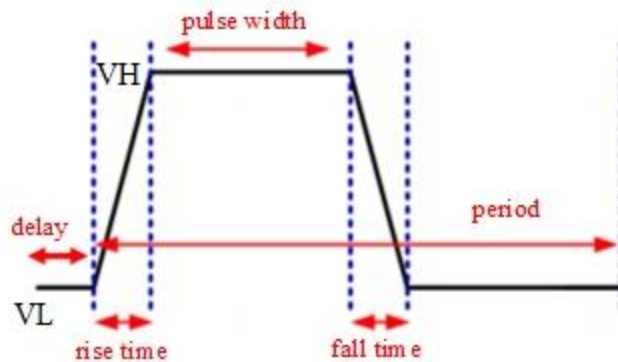
本次作業第一題用到memories觀念，設計6T SRAM cell與周邊電路，並分析其穩定性 - 測量read與write operation時的static noise margin；第二題用到sequencing的觀念，自己決定好測資、要掃的變數值與範圍，畫出latch和flip flop的sequencing element delay，並由圖中分析得setup time等元件參數。

一開始sequencing element delay作圖我跑的圖與同學跑的圖不同，花了不少時間debug，最後找出問題有以下2點

1. 電路接線錯誤，應該是D flip flop輸出點Q再連接4個跟自己一樣的負載(FO4)。

X1	CLK	CLK_b	D	GND	Q	VDD	delay_flip_flop
X2	CLK	CLK_b	Q	GND	output	VDD	delay_flip_flop m = 4

2. 題目要求CLK和CLK_b需使用理想的PULSE，但我CLK和CLK_b給值不同步，參照[HSPICE 入門教學](#)中PULSE的基本語法，如下代碼在一個周期內CLK_b在VL的時間為 $20n - (80p + 80p + 10n) = 10n - 160p$ ，由於CLK與CLK_b互補，故此時間即為CLK在VH的時間。此外CLK與CLK_b的時間差除了 $10p$ ，還要加上CLK_b一開始就會有的rise time $80p$ 。



vclk	CLK	GND	PULSE(0 1.8 '10n+80p' 80p 80p '10n-160p' 20n)
vclk_b	CLK_b	GND	PULSE(0 1.8 0n 80p 80p 10n 20n)