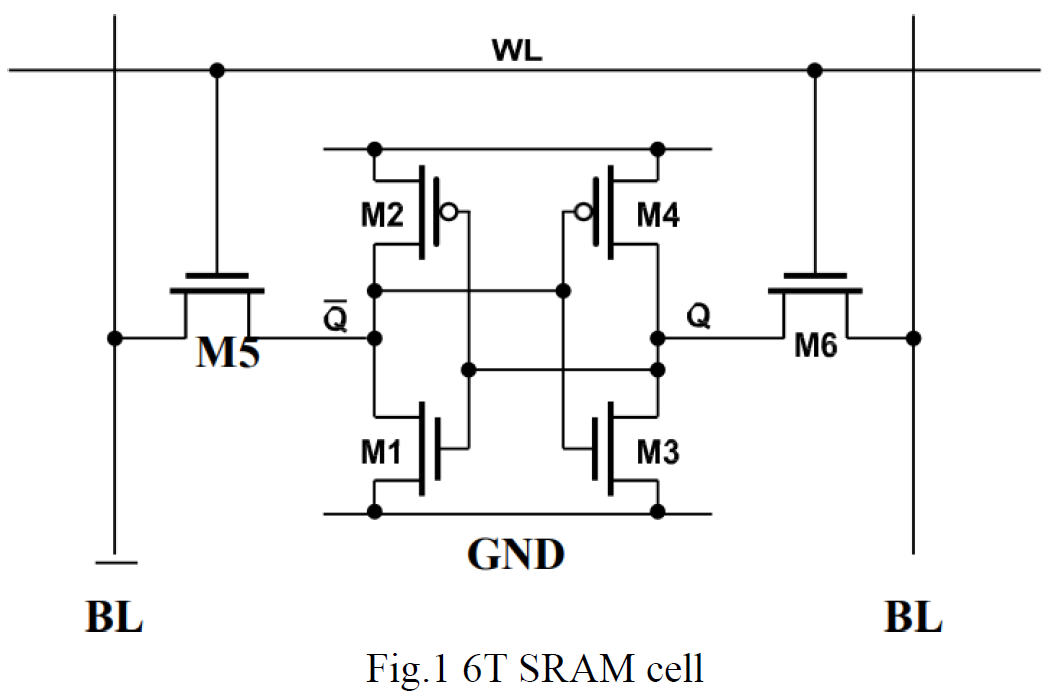
VLSI Circuits Design HW#5

|  |  |  |
| --- | --- | --- |
| 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. 

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。

|  |  |
| --- | --- |
| scratch | schemetic |
|  |  |

分別測量題目要求以下4個pattern

|  |
| --- |
| The stored data(Q) is 0, RE is HIGH |
|  |
| read operation中時bitline conditioning電路的PMOS打開，會先把bit和bit\_b都precharge到，接下來時，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，會把原本precharge到的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，，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。 |

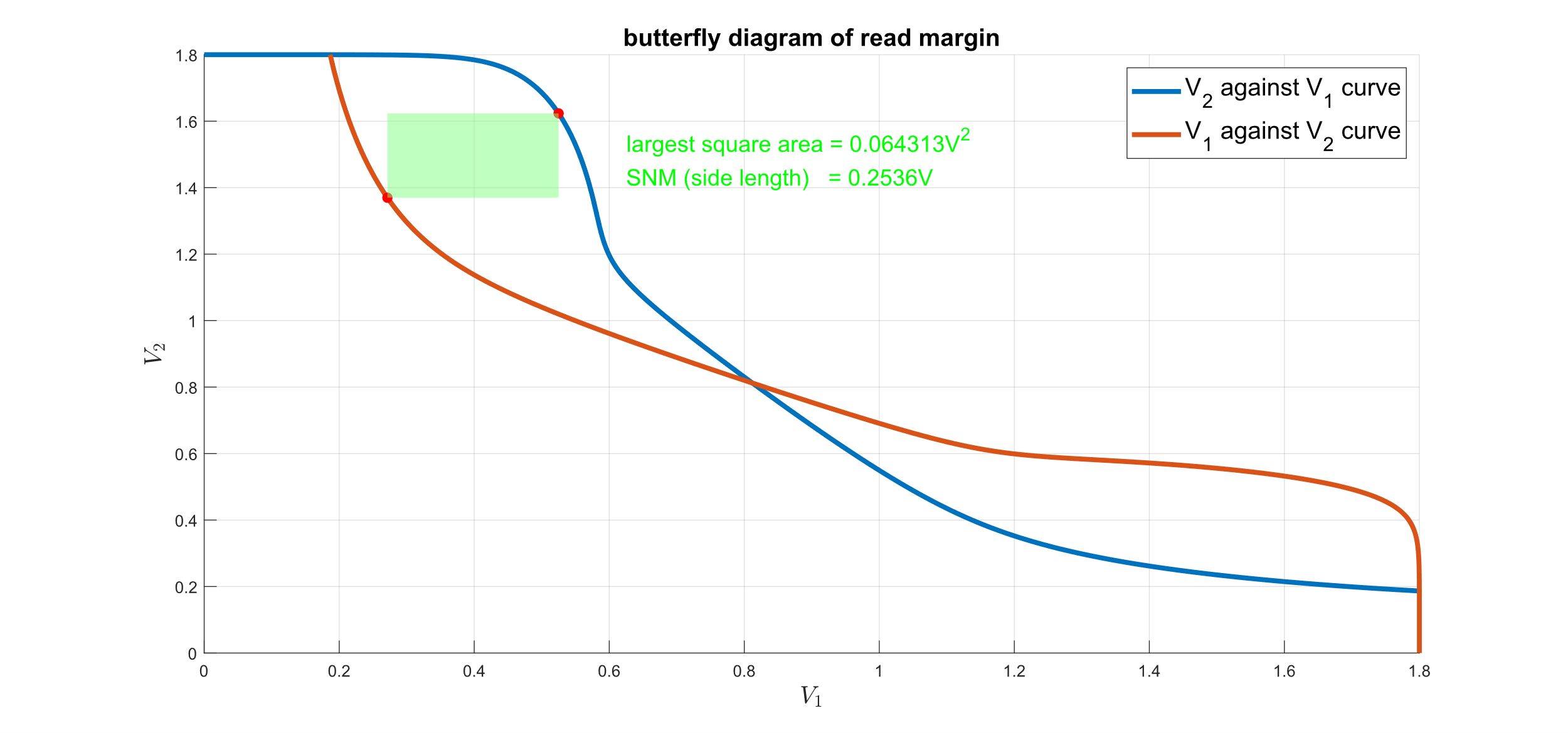
|  |
| --- |
| The stored data(Q) is 1, RE is HIGH. |
|  |
| read operation中時bitline conditioning電路的PMOS打開，會先把bit和bit\_b都precharge到，接下來時，PMOS關閉bit和bit\_b變為floating，再來wordline從0變為1，把N2、N4打開，由於cross-coupled inverter的，會把原本precharge到的bit\_b往下拉至0，代表把data讀出來，因此內部電晶體要比外部讀取電晶體還strong，，此外bit由於floating，會受到wordline從0變為1的耦合電容，因此電壓衝高於。 |
| The stored data(Q) is 0, WE is HIGH, write data is 1. |
|  |
| write operation如同read operation，時bitline conditioning電路的PMOS打開，會先把bit和bit\_b都precharge到，接下來時，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，N1通與N2通，由於前面read operation已經有的條件，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。 |

|  |
| --- |
| The stored data(Q) is 1, WE is HIGH, write data is 0. |
|  |
| write operation如同read operation，時bitline conditioning電路的PMOS打開，會先把bit和bit\_b都precharge到，接下來時，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放電為0，接下來把wordline從0變為1，把N2、N4打開，把bit\_b寫入這個cross-coupled inverter裡面，由於cross-coupled inverter的，N3通與N4通，由於前面read operation已經有的條件，因此寫不進去；改考慮bit寫0時，P1通和N2通，因此設計使N2比P1還strong就可以讓資料寫入cross-coupled inverter裡面，總結條件為。 |

* 1B - read margin of 6T SRAM

DC分析掃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為，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經程式計算即為和下圖二write margin經程式計算即為，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。

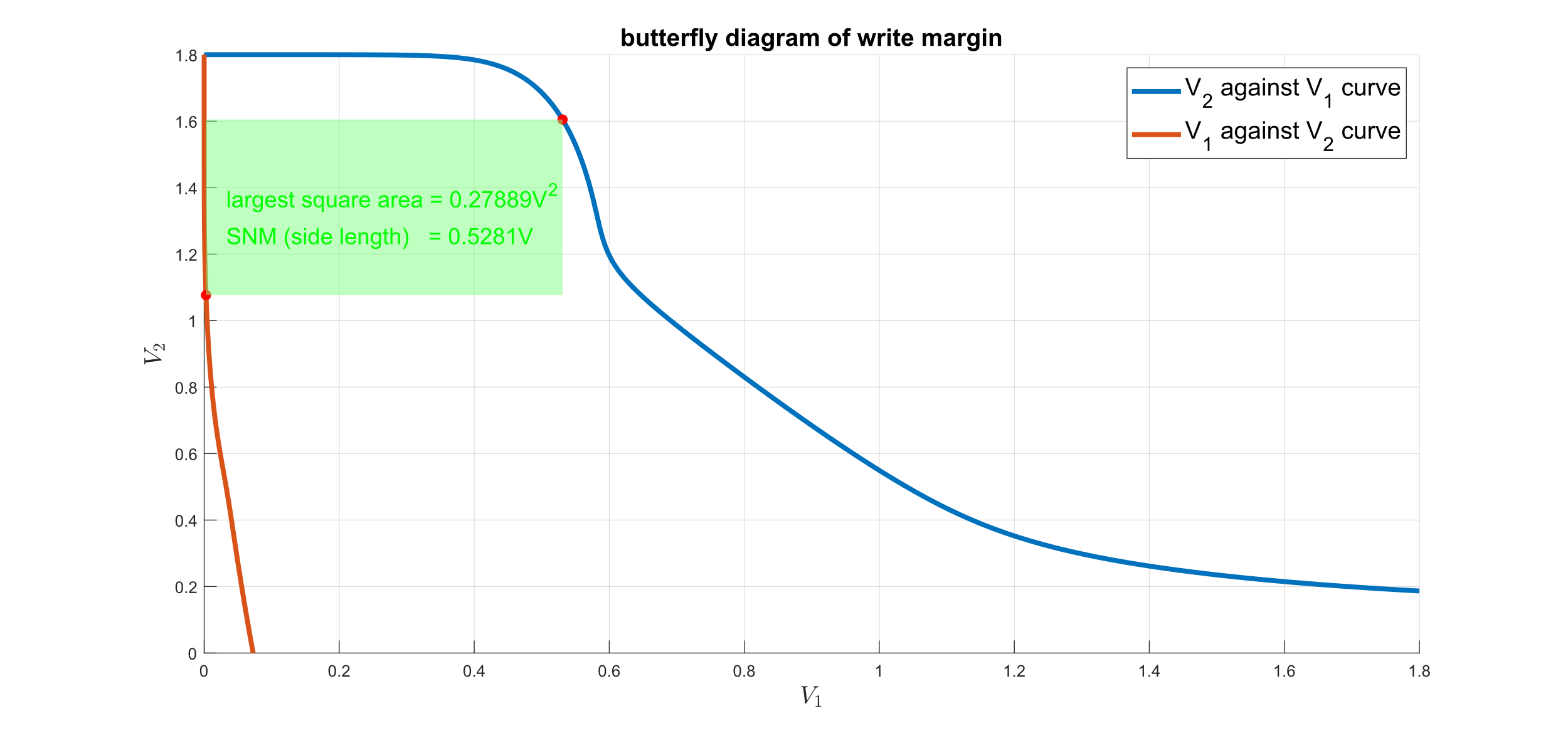


|  |
| --- |
| plot**(**v1**,** v2**,**'LineWidth'**,** 3**);**  plot**(**v2**,** v1**,**'LineWidth'**,** 3**);**  %% find SNM  max\_area **=** 0**;**  index\_point **=** zeros**(**1**,** 2**);**  n **=** length**(**v1**);**  **while** v1**(**n**)** **>** 0.8  k **=** length**(**v1**)** **-** n **+** 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，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** |

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。

|  |  |
| --- | --- |
| 全覽圖 | |
|  | |
| CLK rising edge局部放大圖 | CLK falling edge局部放大圖 |
|  |  |

將\*mt0數據匯入matlab做圖

|  |
| --- |
| 全覽圖 |
| x軸分別就與作圖，可以發現與兩線相交處分別為與，即可推測pulse width為，符合我預先設定的測資。 |
| 局部放大圖 |
|  |

* 2A - find setup time of latch

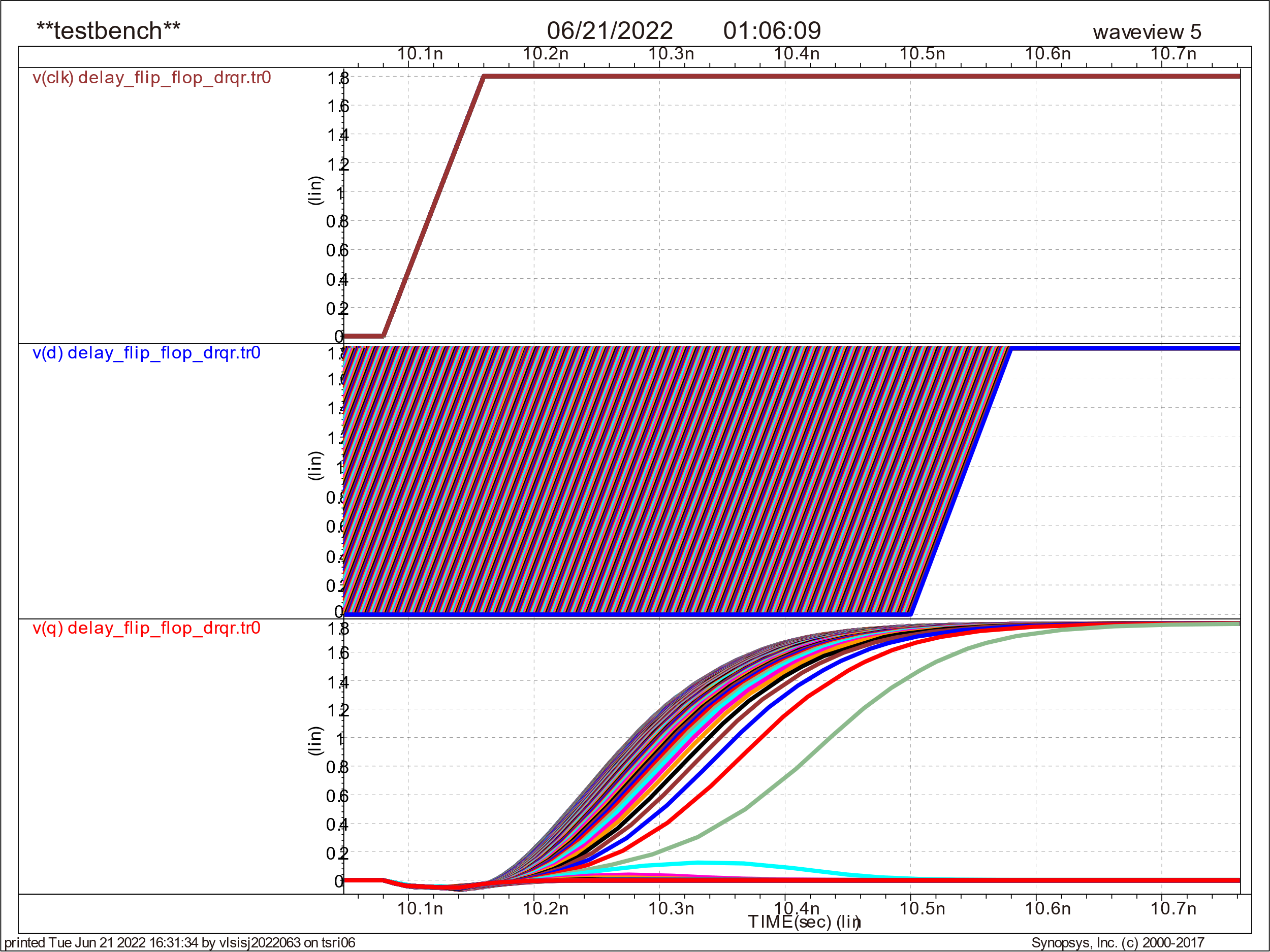
|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

最小值即為，最小值再乘上1.05即為，該點對應的值大小就是。如上圖所示，推測由於latch沒有明顯的transient state，因此隨著下降，沒有delay往上升的趨勢，在小於數值會直接測不到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，不會trigger到此波型，因此沒有測量到此數據而failded，綜合以上原因，我取最後一筆沒有failed的值。

* 2D - sequencing element delay of flip flop

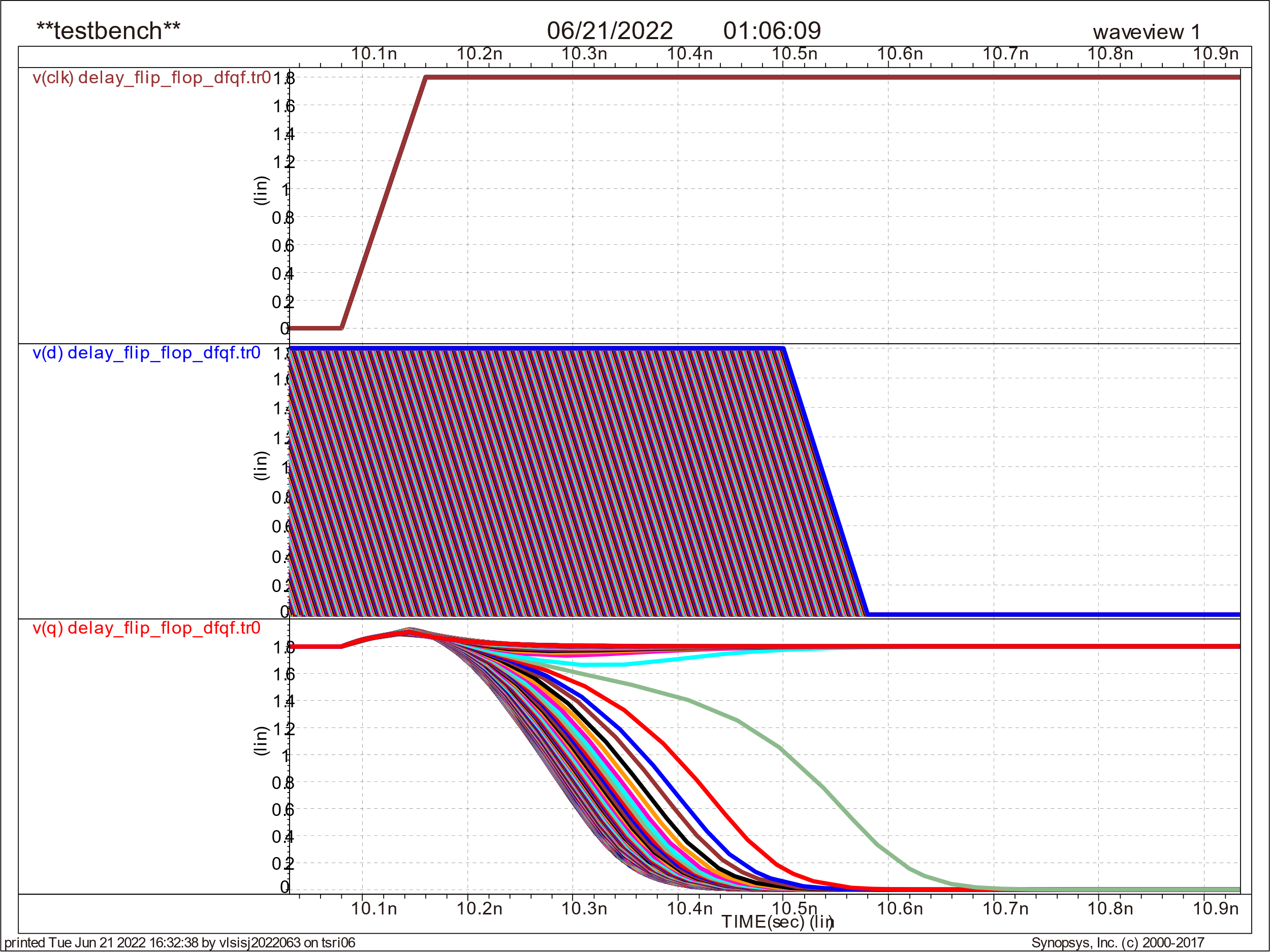
sequencing element delay作圖需要測量4條線，在不同下的延遲時間差，依序為

1. D rises (0 -> 1), Q rises (0 -> 1)測量



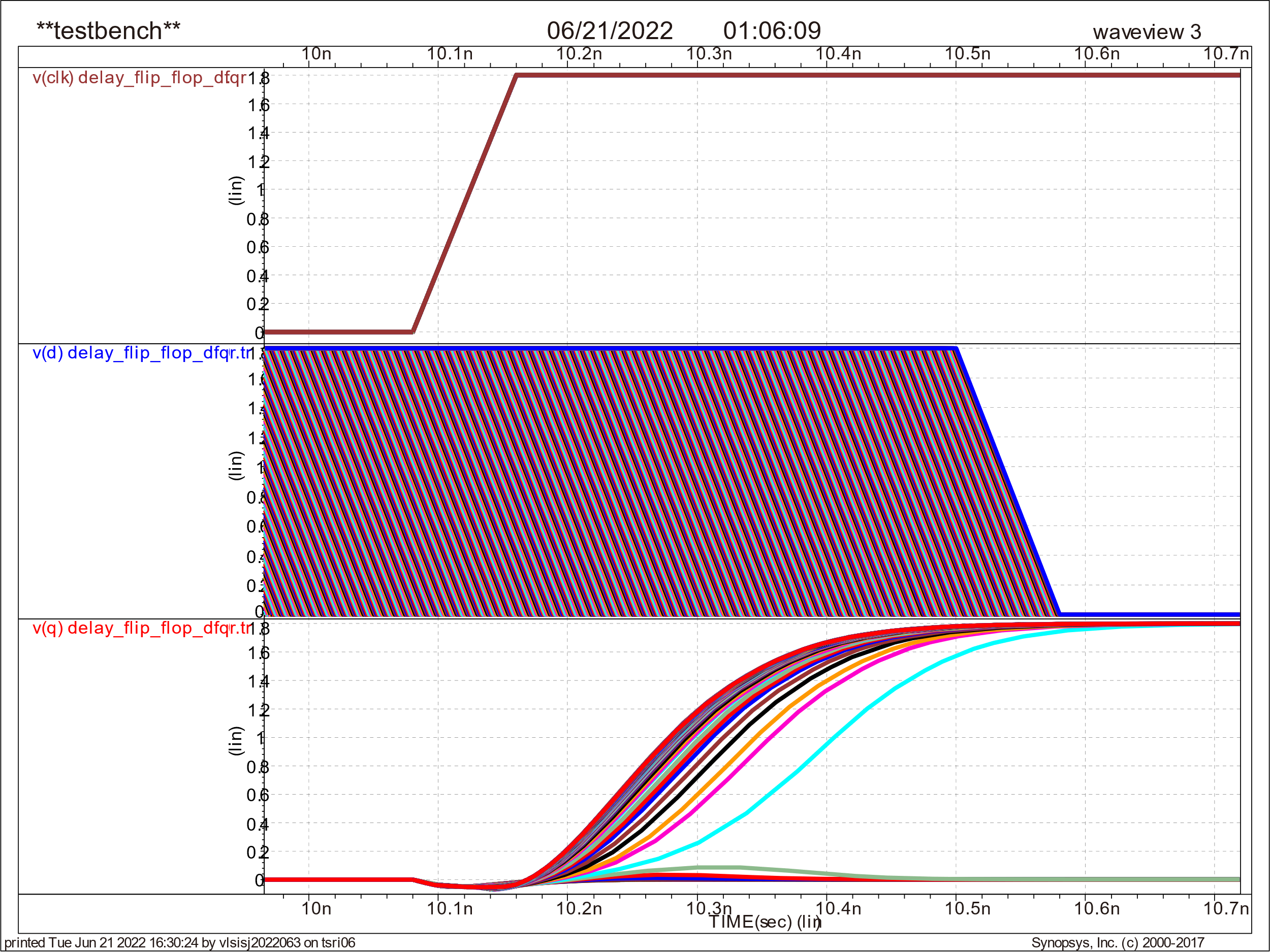
給定Q初始值低電位，在正常情況下CLK positive trigger會抓取輸入D的高電位使得輸出Q電位從設定好的初始值低電位變為高電位，但是隨著減少，D的rising edge往右移，當輸入D小於之後才rise，輸出Q會變成較難轉態，如上圖綠線，繼續減少一直到輸出無法成功轉態而維持在原本的低電位。

1. D falls (1 -> 0), Q falls (1 -> 0)測量



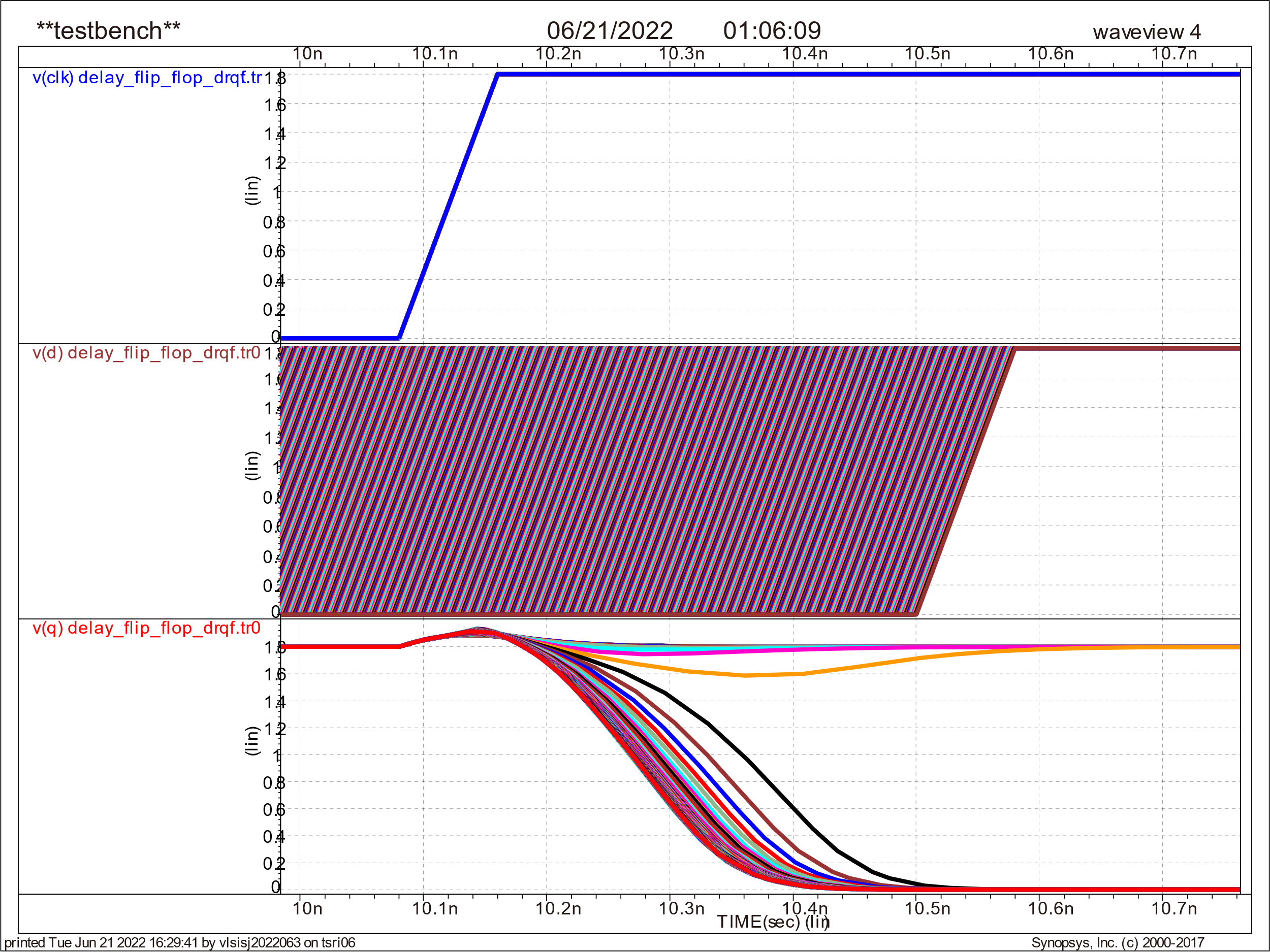
給定Q初始值高電位，在正常情況下CLK positive trigger會抓取輸入D的低電位使得輸出Q電位從設定好的初始值高電位變為低電位，但是隨著減少，D的falling edge往右移，當輸入D小於之後才fall，輸出Q會變成較難轉態，如上圖綠線，繼續減少一直到輸出無法成功轉態而維持在原本的高電位。

1. D falls (1 -> 0), Q rises (0 -> 1)測量



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，但是隨著增加，D的falling edge往左移，當輸入D大於之後才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，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。

1. D rises (0 -> 1), Q falls (1 -> 0)測量



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，但是隨著增加，D的rising edge往左移，當輸入D大於之後才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，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

|  |
| --- |
| 全覽圖 |
|  |

|  |
| --- |
| 局部放大圖 |
|  |

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

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |

在較大處定義，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，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兩條線在斜率為-1時代表在的極(小)值，並於此點定義，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**，並與取距離，距離最近者代表斜率最接近，取此兩數據點的中間值即為要找的。

|  |
| --- |
| 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 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，所以要找離水平線最近的點，該點即為。

有了以上這些數據就可以計算rising input的aperture width 和falling input的aperture width ，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

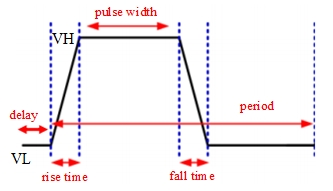
本次作業第一題用到memories觀念，設計6T SRAM cell與周邊電路，並分析其穩定性 - 測量read與write operation時的static noise margin；第二題用到sequecing的觀念，自己決定好測資、要掃的變數值與範圍，畫出latch和flip flop的sequencing element delay，並由圖中分析得setup time等元件參數。

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](https://hackmd.io/@azoo/hspice_tutorial)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，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，還要加上CLK\_b一開始就會有的rise time 。



|  |
| --- |
| 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) |