

設計內容

[1] 設計者姓名與連絡電話

學生姓名: 陳品睿、林沛翰、吳庭毅

聯絡電話: 0974182892 (陳品睿)

Email: b09901194@ntu.edu.tw、b09901192@ntu.edu.tw、b09703086@ntu.edu.tw

[2] 專題名稱

中文專題名稱：帶狀史密斯-沃特曼演算法加速器晶片

英文專題名稱：Banded Smith-Waterman Algorithm Acceleration Chip

[3] 全新設計或改版說明

此案件為改版，設計參考自論文 Adaptively Banded Smith-Waterman Algorithm for Long Reads and Its Hardware Accelerator。

[4] 原理及架構說明

在分子生物學領域中，基因序列配對是一個著名的研究領域。兩基因序列的相似度可以顯現兩物種的親緣關係。如何獲得兩基因的最佳配對方式是一件繁瑣的事情，而現今通常使用如 Smith-Waterman Algorithm 等基因序列配對演算法來加速運算過程。

Smith-Waterman

Algorithm(以下簡稱SW)，是一種著名的基因（核苷酸）序列配對演算法。其運算過程如下：給定兩序列Q（query）和R（reference），以及空位罰分，包含了gap extend, gap open, match, mismatch四種分數。接著，將Q、R分別擺在一矩陣的橫軸以及縱軸，如圖一所示

初始化得分矩陣

	T	G	T	T	A	C	G	G
G	0	0	0	0	0	0	0	0
G	0							
T	0							
T	0							
G	0							
A	0							
C	0							
T	0							
A	0							

圖一

$$D[i, j] = \max \begin{cases} H[i, j-1] - o \\ D[i, j-1] - e \end{cases} \quad (2)$$

$$I[i, j] = \max \begin{cases} H[i-1, j] - o \\ I[i-1, j] - e \end{cases} \quad (3)$$

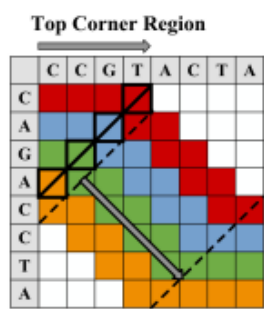
$$H[i, j] = \max \begin{cases} 0 \\ I[i, j] \\ D[i, j] \\ H[i-1, j-1] + \delta(r_j, q_i) \end{cases} \quad (4)$$

$$T[i-1, j-1] = \arg \max \begin{cases} 0 \\ I[i, j] \\ D[i, j] \\ H[i-1, j-1] + \delta(r_j, q_i) \end{cases} \quad (5)$$

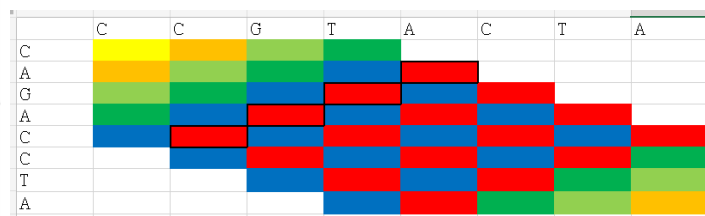
圖二

原始的SW會由左至右、由上至下依照圖二的規則計算每一個格子的比對分數，其中，**H**為分數矩陣（儲存分數），**T**為回朔矩陣（儲存繼承哪個方位的路徑）。最後，再從分數最高的地方照回朔矩陣的路徑回朔，並將遇到的配對方式儲存起來並輸出。

然而，根據經驗與實際情況，會被拿來比對的兩基因本身就具有一定的相似度，因此最高分的配對方式通常會出現在矩陣的帶狀區域。因此，此篇論文作者提出了一個改良版的配對演算法—**adaply Banded Smith-Waterman Algorithm**。這個演算法只會搜尋矩陣的帶狀區域，以此來減少資料儲存空間以及減少運算時間同時又能獲得不錯的結果。以圖三為例，只有著色部分會被計算。

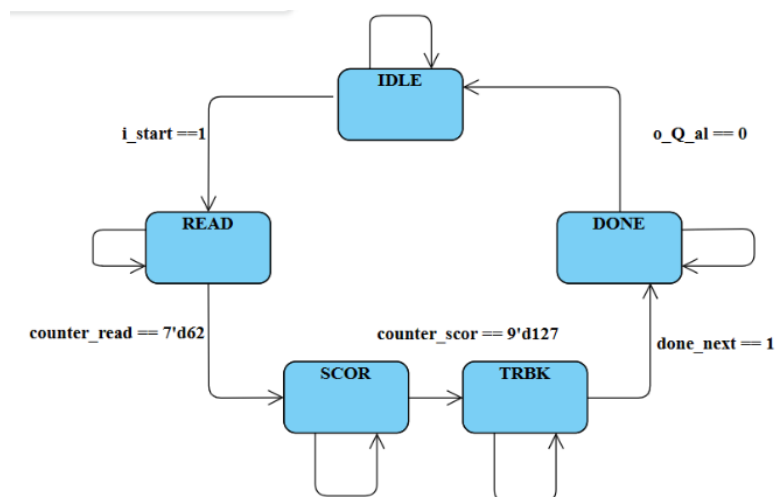


圖三



圖四

圖五是有限狀態機。主要分成讀取、計算分數即回朔三個部分。

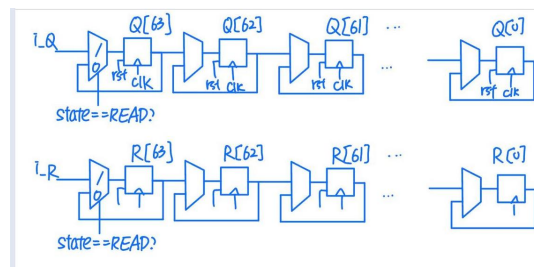


圖五

首先收到start訊號後會進入讀取狀態，讀取要被配對的兩條基因序列。讀取完後，會進入計算分數狀態，從左上一路沿著對角線帶狀區域算到右下角。計算完後，會從分數最高的地方回朔，並同時儲存配對結果。等到已經無法再繼續回朔時，就會開始讀取配對結果並輸出至外部文件。

（ 1 ） READ-Phase

首先，要先讀取要被配對的兩基因序列。受限於腳位數量，一次只能讀取一對基因，因此使用近似於shift register(with enable)的方式讀取資料。如圖六



圖六

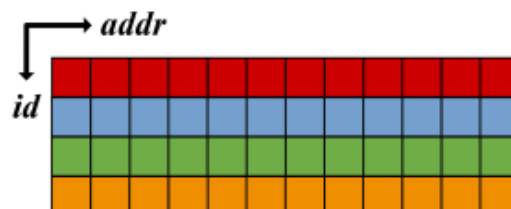
(2) SCOR-Phase

接著，要計算帶狀區域每一格的分數並同時儲存回溯時所需的相對位置資訊。

首先，SCOR-Phase

的運作過程為如漣漪般由左上角至右下角計算，圖三、圖四為示意圖。圖三中黑色框框部分會在同一個CYCLE同時運算，下一個cycle會往右下角下一排移動。

接著，因為運算區域為帶狀區域，且斜狀區域固定寬度為6（圖例為寬度4），所以要用比較特殊的方法儲存相對位置的資訊，也就是圖七id, addr
的方法。圖七的紅色register
file儲存左上圖紅色區域的格子，其他顏色依此類推。而至於要存在第幾個addr(e.g. 這個id的第幾個位置)，則依據當下在第幾個cycle與哪一個id決定。

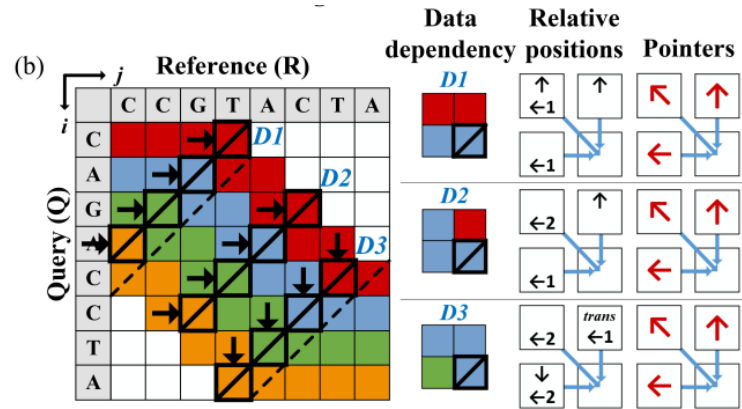


圖七：register file storing relative positions

至於要存什麼資訊進register

file，從圖八可以發現運算過程中對於每一個方塊可以大致分成3種情況。

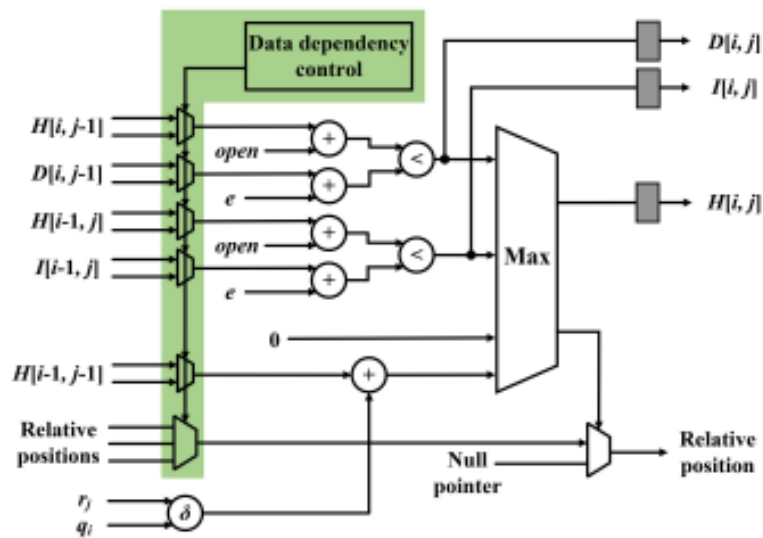
- （一）左上與右下區域。可以發現此區域的方塊的繼承情況屬於D1。
- （二）帶狀的兩種區域：D2和D3交錯出現。



圖八：相對位置是如何存進暫存器堆

對於每個方塊會有三種可能繼承方位：上方、左上、左邊。以D1為例，若從上方繼承，表示前一個的位置應該在上一個id、同一個addr；若是從左上繼承，則會是上一個id、前一個addr；若是從左邊繼承，則是同一個id、前一個addr。D2和D3可以用類似方法得知（同色為同一排id），向上箭頭即是前一個id；向左箭頭和2則是前兩個addr。雖然總共有9種可能，但扣除重複只會有6種，所以暫存器為每3個bit為一組。

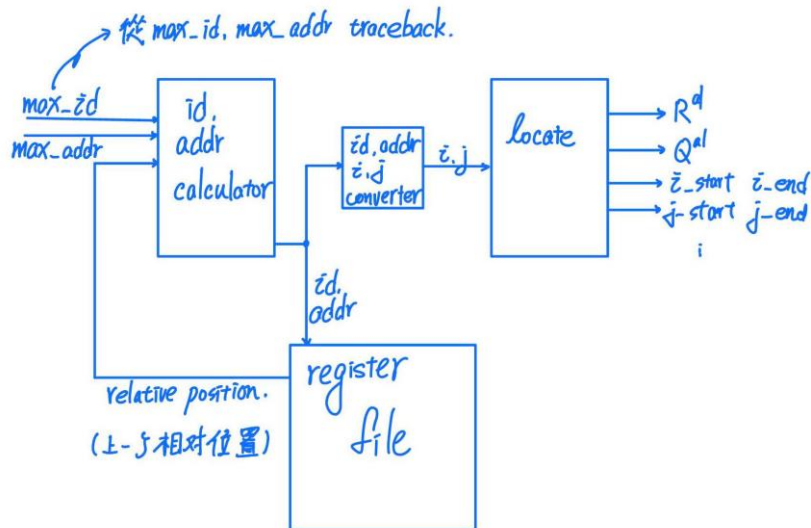
分數計算方式是依照圖二的方式合成，稱之為PE，合成結果如圖九。並且為了平行化，PE數量會等於帶狀寬度。最後再把每個PE產生的score與之前的最大值比較並儲存最大值與最大值的位置。



圖九：PE邏輯圖

(3) TRBK-Phase

最後，根據儲存下來的相對位置資訊，從分數最大值的方塊回溯直到遇到邊界或是分數為0，而兩基因的最佳匹配分數也就是該分數最大值。回溯模組的邏輯圖如圖十：



圖十：回溯模組邏輯圖

在每一個cycle，回溯模組同時會做兩件事：輸出當下排列後的Q和R應儲存的內容以及找在下一個cycle id, addr的位置。在每個cycle的當下，回溯模組會有一筆現在位於的id和addr，其中，id是指第幾個PE；addr是指在PE的第幾個位置。id, addr會經過公式轉換得到在矩陣的座標，並依據此座標以及相對位置決定輸出的Q和R（Q的第x個位置或是空一格）。除此之外，回溯模組也會依據從register file收到的相對位置資訊計算下一個（或應該稱之更靠近頭的）id, addr，並在下一個cycle的開始將值送給回溯模組。一直重複這兩個階段直到無法再回溯為止。

（4）比較結果

a.突變率比較

測試不同突變率對正確率的影響，帶狀寬度為4。可以發現loss都非常低。但為了百分之百正確率最後選擇寬度為6作為最終定案。（寬度6可以完全正確）

測資：大小20 * 1000k組

Mutation rate		SW(C++)	BSW(C++)	loss
30%	Execution time(s)	24.1	14.28	
	Total score	26508915	26496665	0.046%

40%	Execution time(s)	22.5	13.61	
	Total score	24914017	24888953	0.1006%

表一：突變率比較

b.帶狀寬度比較

10000組測資，clock cycle= 100MHz。帶狀寬度不影響時間，只影響面積

最終選擇case2作為最終結果

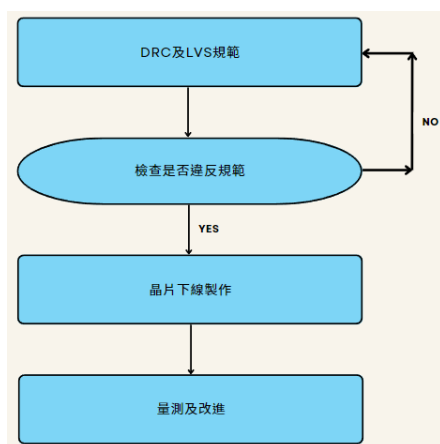
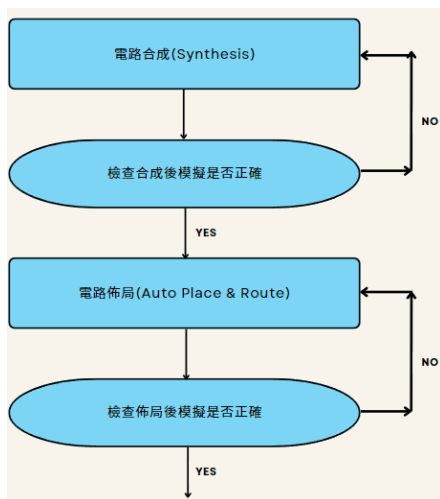
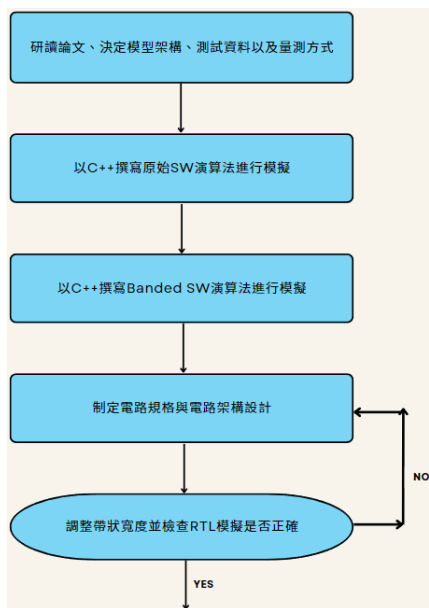
	Read length	Band width	area	Time(RTL)	Time(synthesis)
Case 1	64	4	355874	32,205,355ns	32,205,355ns
Case 2	64	6	468810	32,205,355ns	32,205,355ns
Case 3	64	8	568916	32,205,355ns	32,205,355ns

表二：帶狀寬度比較

Signal Name	I/O	Width	Function
clk	I	1	clock
rst	I	1	reset
start	I	1	activate algorithm
i_Q	I	3	base of input Q
i_R	I	3	base of input R
o_Q_al	O	3	base of output Q
o_R_al	O	3	base of output R
detail_info	O	9	information of i/j start/end and score
done	O	1	activate output checking

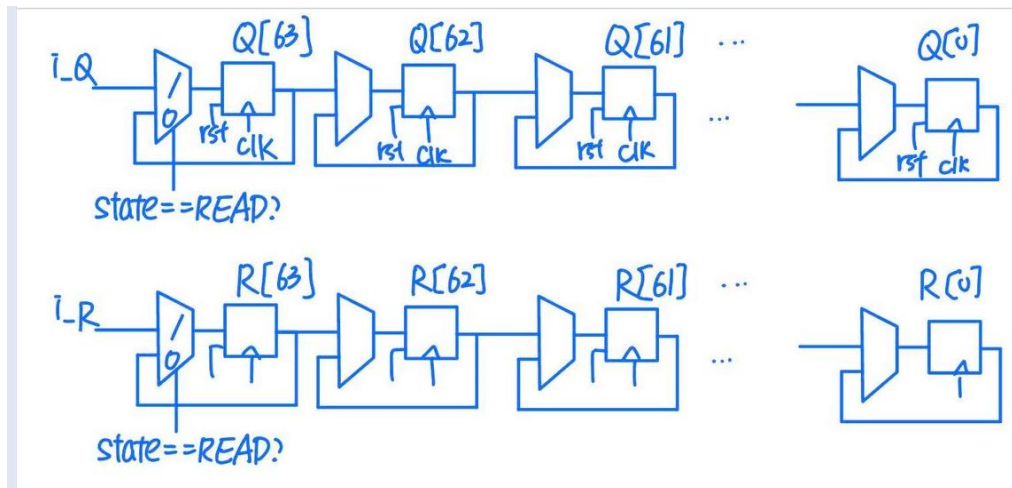
表三 輸入及輸出腳位

[5] 設計流程

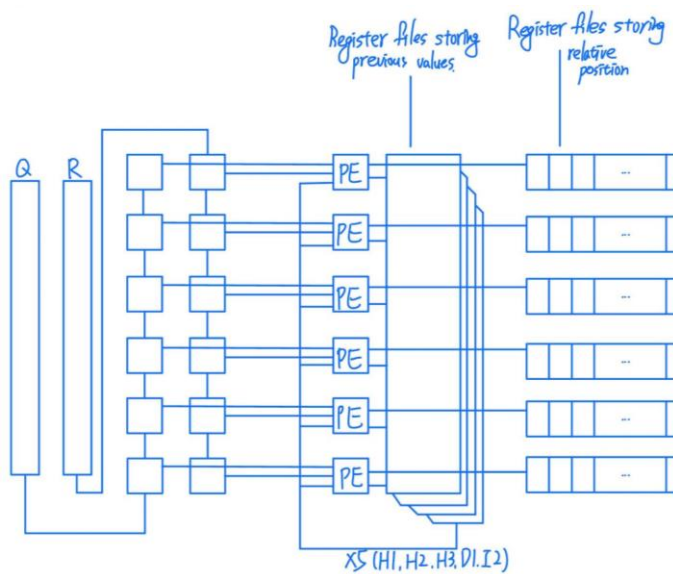


[6] 電路詳圖

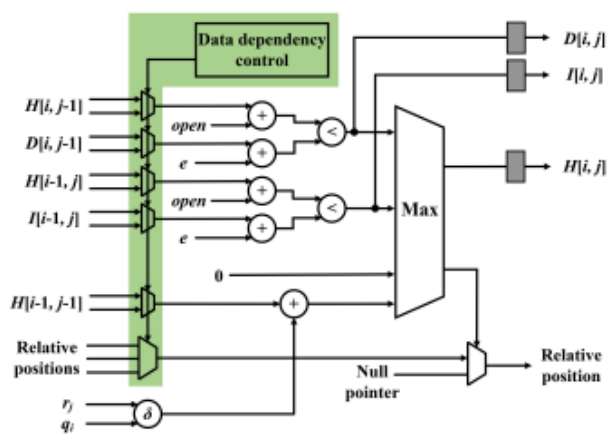
(一) Read Phase



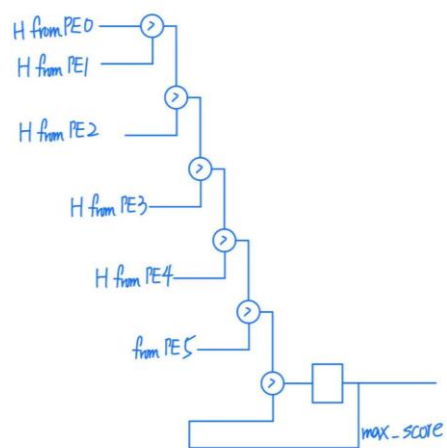
(二) scoring phase



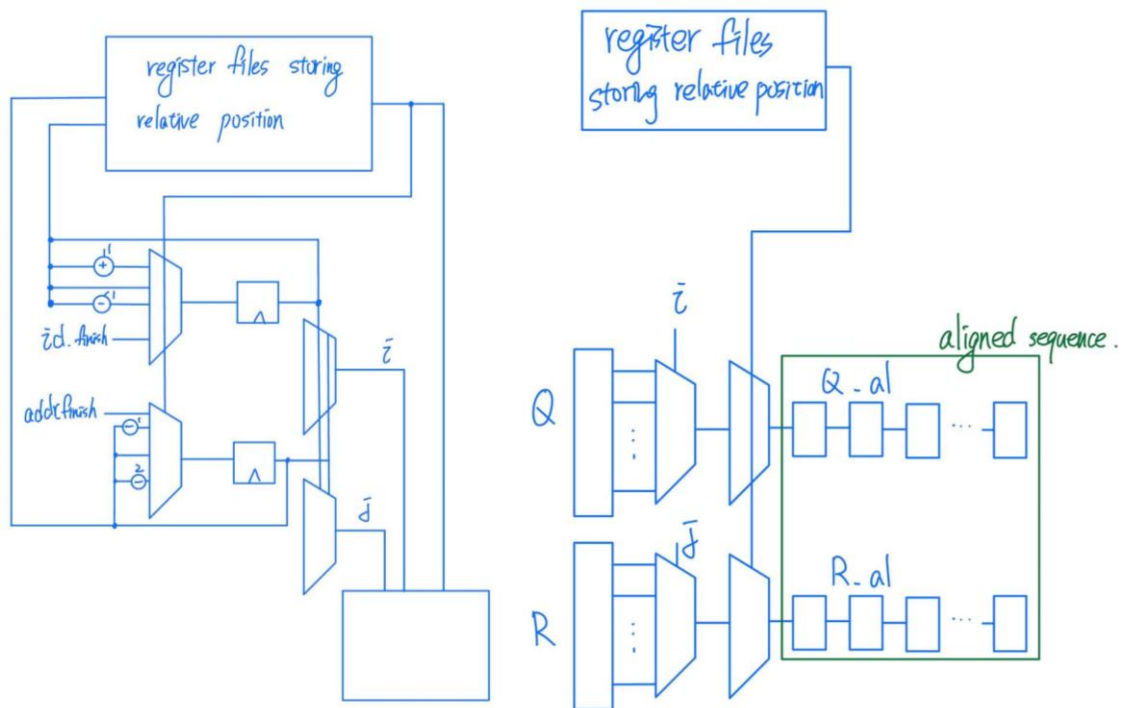
PE



new_Max_score



(三) Traceback



[7] 模擬結果

1. Pre-Layout Simulation

將Pre-Layout產生的匹配方式與C++產生的匹配方式分別輸出到兩個文件檔並用程式比較兩文件檔是否完全一樣（包含基因匹配方式、輸出的Q和輸出的R的起始、結束位置在輸入Q和輸入的R的位置以及匹配分數）。由圖片可知他們內容完全一樣。

ATCTCTGCAGCGCATGAATCGGAACGCTGCCGATTCCATCGGTTACAACCGCTGCGCA	ATCTCTGCAGCGCATGAATCGGAACGCTGCCGATTCCATCGGTTACAACCGCTGCGCA
ACCGTC	ACCGTC
ATCTCTGCTGCGCATGAATCGGAACGTTGCCGATTCCATCGATTACAACCGCGGCGCA	ATCTCTGCTGCGCATGAATCGGAACGTTGCCGATTCCATCGATTACAACCGCGGCGCA
ACCGTC	ACCGTC
0	0
0	0
63	63
63	63
116	116
ACAATACATTGGGTAACTTTTCAGGATTAATCATGAGGCTGTAACACACGTGCTGTTA	ACAATACATTGGGTAACTTTTCAGGATTAATCATGAGGCTGTAACACACGTGCTGTTA
GCAAA	GCAAA
ACAATACATTGGGTAACTTTTCGGGATTAATCATGAGGCTGTAACACGCGTGCTGTTA	ACAATACATTGGGTAACTTTTCGGGATTAATCATGAGGCTGTAACACGCGTGCTGTTA
GCAAA	GCAAA
0	0
0	0
63	63
63	63
122	122
ATACGCTGCGTGATGGAACCATGCTAGTACATGAGCCATAGGGGACCACTCCCCAGCA	ATACGCTGCGTGATGGAACCATGCTAGTACATGAGCCATAGGGGACCACTCCCCAGCA
ACCGT	ACCGT
ATACGCTGCGTGATGGAACCATGCTAGTACATGAGCCATAGGGGACCACTACCAAGCA	ATACGCTGCGTGATGGAACCATGCTAGTACATGAGCCATAGGGGACCACTACCAAGCA
ACCA-T	ACCA-T
0	0
0	0
63	63
62	62
122	122
ATAGGAAGACAAGTAACAATCCTCAAGGTACTGTTGTTTTACTATTTTCAGGTCGCT	ATAGGAAGACAAGTAACAATCCTCAAGGTACTGTTGTTTTACTATTTTCAGGTCGCT
CGCCC	CGCCC
ATAGGAGGACAAGTAAAAATCCATCAAGGTTACTGTGTTCTTACTATTTTCAGGTCGCT	ATAGGAGGACAAGTAAAAATCCATCAAGGTTACTGTGTTCTTACTATTTTCAGGTCGCT
CGCCC	CGCCC
0	0
0	0
63	63
63	63
113	113
CTAGAGCCTGTGGGCGCCAAGCACCCCTATGTTCTAAGTGAACGCTTACCGACCCAA	CTAGAGCCTGTGGGCGCCAAGCACCCCTATGTTCTAAGTGAACGCTTACCGACCCAA
TTACAT	TTACAT
CTAGAGCCTGTGGGCGCCAAGCACCCCATATGTTATGAGTCAACACCTACCGACCCAA	CTAGAGCCTGTGGGCGCCAAGCACCCCATATGTTATGAGTCAACACCTACCGACCCAA
TTATAT	TTATAT
0	0
0	0
63	63
63	63
107	107

2. Post-Layout Simulation

將Pre-Layout產生的匹配方式與C++產生的匹配方式分別輸出到兩個文件檔並用程式比較兩文件檔是否完全一樣（包含基因匹配方式、輸出的Q和輸出的R的起始、結束位置在輸入Q和輸入的R的位置以及匹配分數）。由圖片可知他們內容完全一樣。

ATCTCTGCAGCGCATGAATCGGAACGCTGCCGATTCCATCGGTTACAACCGCTGCGCA	ATCTCTGCAGCGCATGAATCGGAACGCTGCCGATTCCATCGGTTACAACCGCTGCGCA
ACCGTC	ACCGTC
ATCTCTGCTGCGCATGAATCGGAACGTTGCCGATTCCATCGATTACAACCGCGGCGCA	ATCTCTGCTGCGCATGAATCGGAACGTTGCCGATTCCATCGATTACAACCGCGGCGCA
ACCGTC	ACCGTC
0	0
0	0
63	63
63	63
116	116
ACAATACATTGGGTAACTTTTCAGGATTAATCATGAGGCTGTAACACACGTGCTGTTA	ACAATACATTGGGTAACTTTTCAGGATTAATCATGAGGCTGTAACACACGTGCTGTTA
GCAAA	GCAAA
ACAATACATTGGGTAACTTTTCAGGATTAATCATGAGGCTGTAACACACGTGCTGTTA	ACAATACATTGGGTAACTTTTCAGGATTAATCATGAGGCTGTAACACACGTGCTGTTA
GCAAA	GCAAA
0	0
0	0
63	63
63	63
122	122
ATACGCTGCGTGATGGAACCATGCTAGTACATGAGCCATAGGGGACCACTCCCCAGCA	ATACGCTGCGTGATGGAACCATGCTAGTACATGAGCCATAGGGGACCACTCCCCAGCA
ACCA-T	ACCA-T
0	0
0	0
63	63
62	62
122	122
ATAGGAAGACAAGTAACAATCCTTCAAGGTTACTGTTGTTTTTACTATTTTCAGGTCGCT	ATAGGAAGACAAGTAACAATCCTTCAAGGTTACTGTTGTTTTTACTATTTTCAGGTCGCT
CGCCC	CGCCC
ATAGGAGGACAAGTAAAAATCCATCAAGGTTACTGTCGTTCTTACTATTTTCAGGTCGCT	ATAGGAGGACAAGTAAAAATCCATCAAGGTTACTGTCGTTCTTACTATTTTCAGGTCGCT
CGCCC	CGCCC
0	0
0	0
63	63
63	63
113	113
CTAGAGCCTGTCGGGCGCAAGCACCCCTATGTTCTAAGTGAACGCTTACCGACCCAA	CTAGAGCCTGTCGGGCGCAAGCACCCCTATGTTCTAAGTGAACGCTTACCGACCCAA
TTACAT	TTACAT
CTAGAGCCTGTCGGGCGCAAGCACCCCTATGTTCTAAGTGAACGCTTACCGACCCAA	CTAGAGCCTGTCGGGCGCAAGCACCCCTATGTTCTAAGTGAACGCTTACCGACCCAA
TTATAT	TTATAT
0	0
0	0
63	63
63	63
107	107

[8] 量測流程

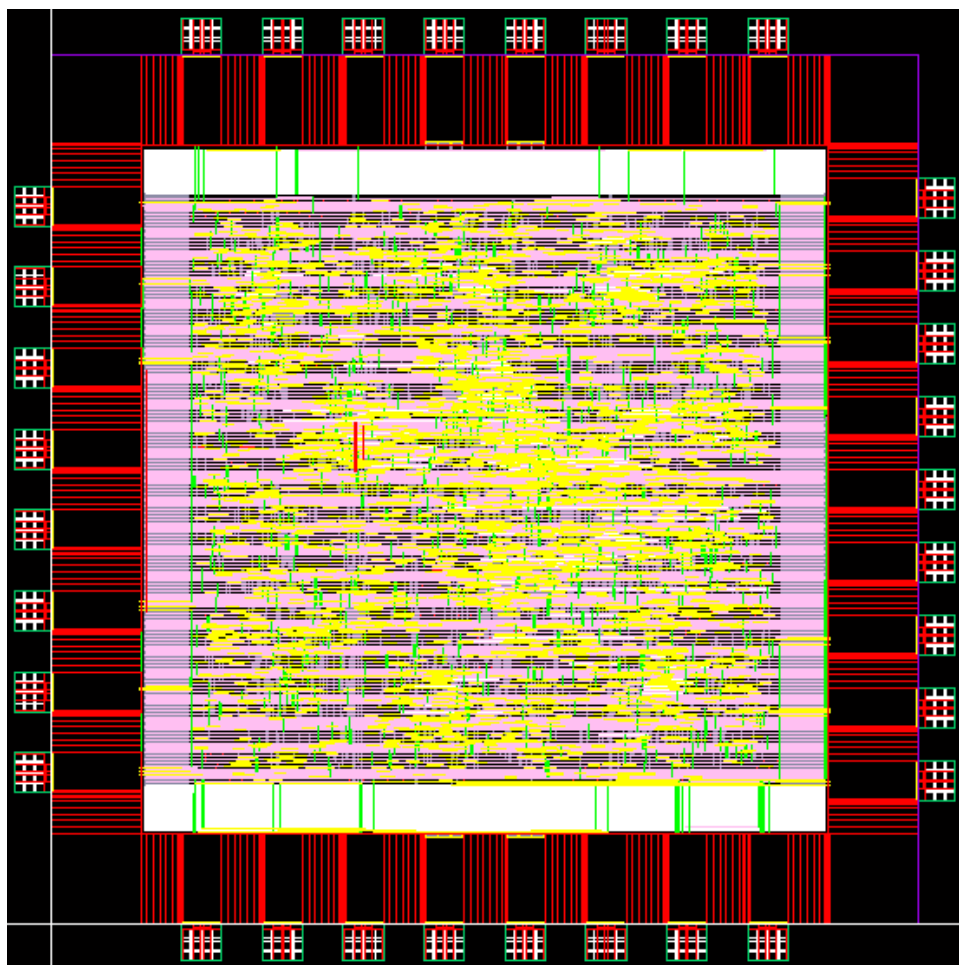
1. 先產生多筆基因序列，每筆基因序列為一條 R (reference)。
2. 接著設置讓每一個鹼基產生突變（鹼基改變）的突變率並應用在每一個R的鹼基。令每一條R突變後的結果為Q(query)，接著再把R與對應的Q配對並計算分數。
3. 利用C++模擬SW algorithm 與 Banded-SW algorithm電路，並將input、output萃取出來，即為測試資料。
4. 儀器設置：電源供應器調整為3.3V直流電，並接上晶片的電源腳位
5. 儀器設置：訊號產生器接至晶片的輸入腳位，並產生100MHz的方波，輸入clk腳位
6. 儀器設置：將邏輯分析儀接至晶片的輸出腳位
7. 使用訊號產生器向晶片輸入所需的input和output測試資料
8. 將晶片輸出與3.的output資料比對，確認是否與模擬結果相當

DRC 驗證結果：皆為可容許之假錯

LVS驗證結果：通過

```
#####  
##                               ##  
##      CALIBRE SYSTEM          ##  
##                               ##  
##      LVS REPORT              ##  
##                               ##  
#####  
  
REPORT FILE NAME:    lvs_test.rep  
LAYOUT NAME:        svdb/CHIP.sp ('CHIP')  
SOURCE NAME:        CHIP.spi ('CHIP')  
RULE FILE:          G-DF-MIXED_MODE_RFCMOS18-1.8V_3.3V-1P6M-NMC_CALIBRE-LVS-2.1-P8.txt  
RULE FILE TITLE:     LVS of UMC 0.18um 1.8V/3.3V 1P6M NMC Mixed Mode/RFCMOS Process  
HCELL FILE:         (-automatch)  
CREATION TIME:       Fri May 26 06:21:10 2023  
CURRENT DIRECTORY:   /home/raid7_2/user/b09/b09194/ICDE_Final/Lab5_LVS/Lab5_LVS  
USER NAME:           b09194  
CALIBRE VERSION:     v2022.3_33.19 Tue Sep 6 12:10:05 PDT 2022  
  
  
OVERALL COMPARISON RESULTS  
  
# # # # #  
# # CORRECT #  
# # # # #  
# #  
  
-----  
CELL SUMMARY  
-----  
  
Result Layout Source  
-----  
CORRECT CHIP CHIP
```

[10] 佈局平面圖

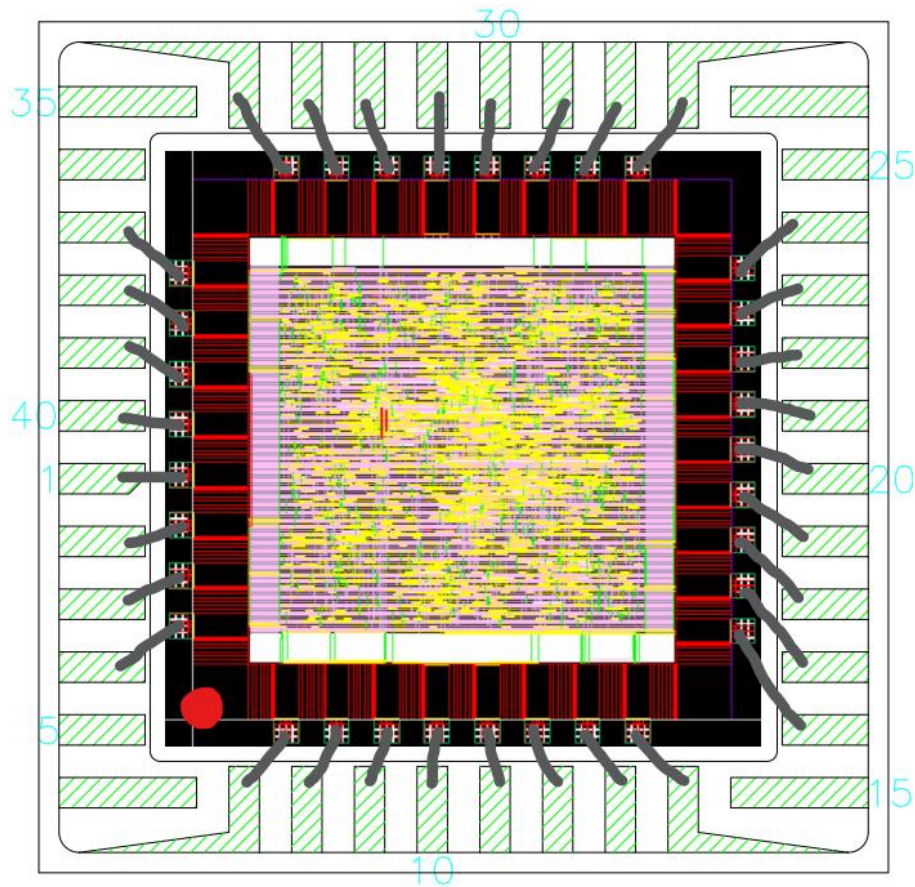


Chip Size: 1464.00mm * 1463.58mm

Power Dissipation: 13.2694 mW

Max Frequency: 71.429MHz (1/(14ns))

[11] 打線圖



[12] 預計規格列表

Description	
Process	UMC 0.18um Mixed-Mode and RFCMOS 1.8V/3.3V 1P6M Metal Metal Capacitor Process
Power Supply	1.8V

Specification	Spec.	Pre-sim	Post-sim
Frequency	71.429 MHz	71.429 MHz	71.429 MHz
Chip size	< 1.5 mm*1.5 mm	455502 μm^2	1464.00mm *

			1463.58mm
Power	-	13.2694 mW	13.2694 mW
PADs	33	33	33

[13] 參考文獻

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8445105>