

# 2022 IC Design Contest

## Cell-Based IC Design Category for Undergraduate Level

### *Burrows Wheeler Transform Engine*

#### 1. 問題描述

請完成 Burrows Wheeler Transform Engine(後文以 **BWTE** 表示)功能的電路設計。本電路可將自測試輸入字串依需要轉換成對應的 BWT 字串並對一短字串進行比對定位。；、：【】

透過 BWTE 使相同字元位置連續或相鄰，以加速後續其他演算法對特定字串片段進行搜尋與比對的效率。過去被用於搜尋引擎來搜尋任意大小的字串片段，如今則被廣泛用於次世代基因定序工具中，面對動輒以 TB 論的龐大基因序列，業界已將 BWTE IC 化以加速特定基因片段的比對工作，可應用在胎兒基因疾病變異株搜尋、病毒株溯源等基因工程，藉由硬體化可改善至少 50 倍的處理能力。

有關 BWTE 詳細規格將描述於後。表一為本電路各輸入、輸出信號的功能說明。各參賽隊伍必須依照下一節所指定的設計規格及附錄 B 中的測試樣本完成設計驗證。

本次 IC 設計競賽比賽時間為上午 08:30 到下午 20:30。當 IC 設計競賽結束後，本題會根據第三節中的評分標準進行評分。為了評分作業的方便，各參賽隊伍應參考附錄 C 與附錄 D 中所列的要求，附上評分所需要的檔案。

本題目之測試樣本置於 **`/usr/cad/icc2022/euc/icc2022euc.tar`**，請執行以下指令取得測試樣本：

**`tar xvf /usr/cad/icc2021/euc/icc2022euc.tar`**

**軟體環境及設計資料庫說明請參考附錄 E 與附錄 F。**

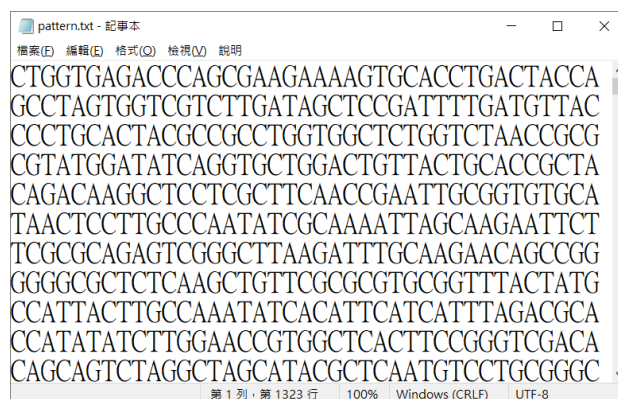


圖 1、基因序列示意圖

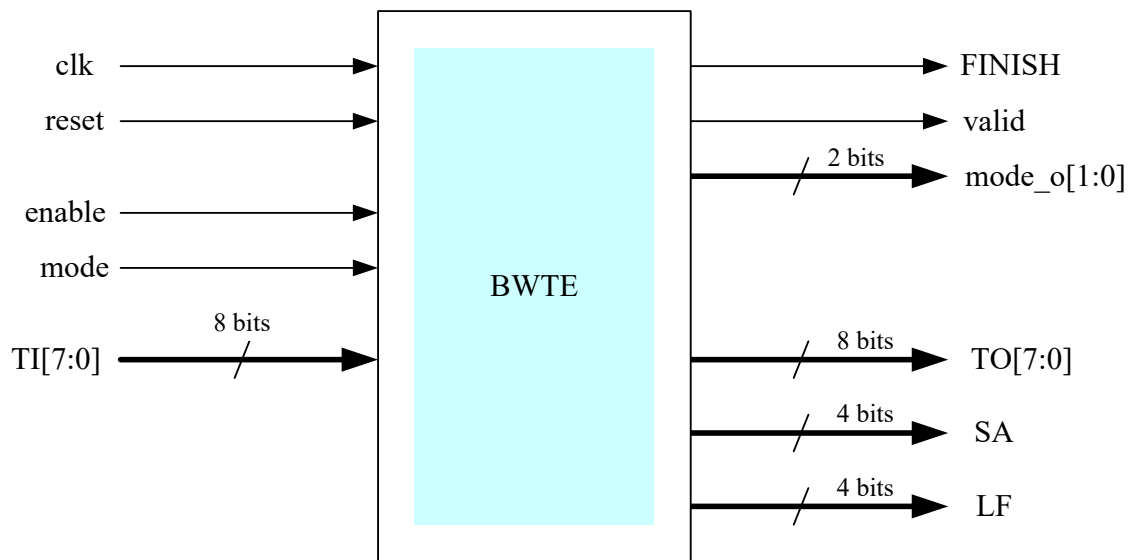


圖 2、Burrows Wheeler Transform Engine 之方塊圖

# 2.設計規格

## 2.1 系統方塊圖

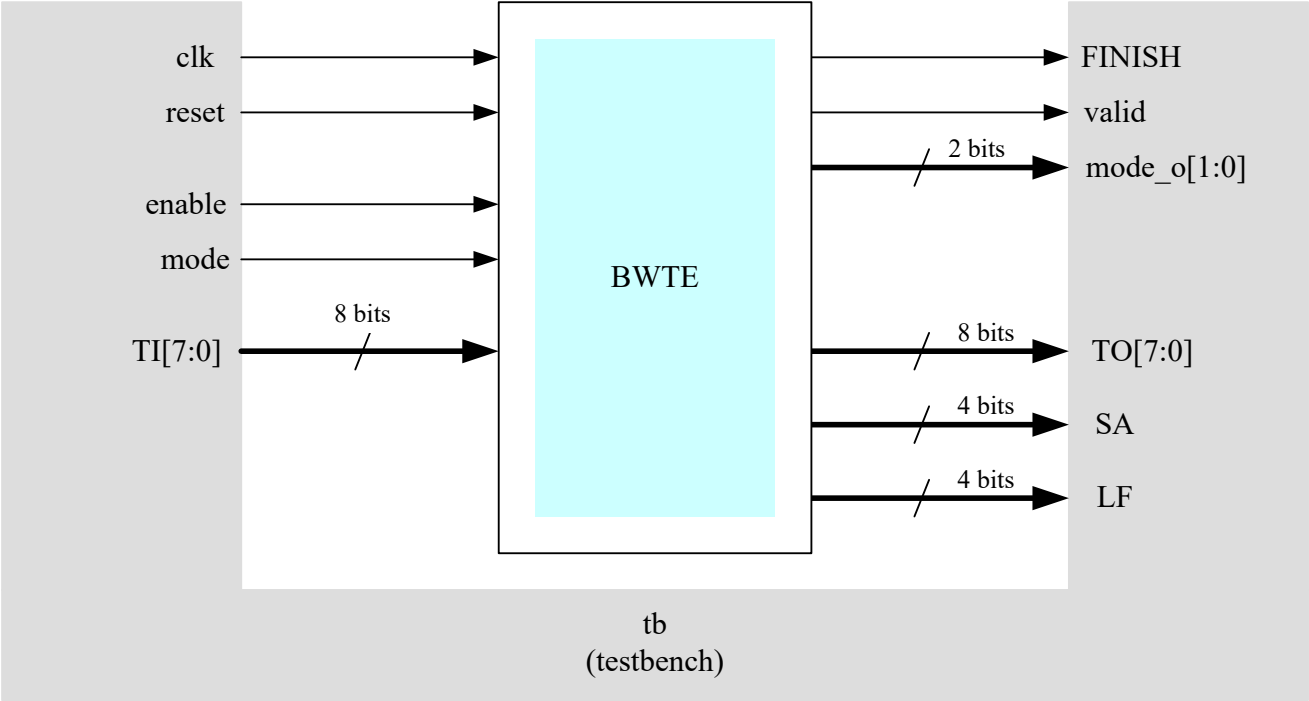


圖 3、Burrows Wheeler Transform Engine 之系統方塊圖

## 2.2 輸入/輸出介面

表 1 -輸入/輸出訊號

Signal Name	I/O	Width	Simple Description
clk	I	1	系統工作時脈，本系統為同步於時脈正緣之同步設計。 (註: Host 端採 clk 正緣時存取資料。)
reset	I	1	系統重置訊號 Reset Signal (active high)。由 testbench 提供，拉高 2 cycle 後恢復為 low。
enable	I	1	輸入致能(enable)控制訊號，當 enable 為 High (1)時，輸入訊號 TI 方為有效。
mode	I	1	輸入資料模式控制訊號。當 enable 為 High 時，mode 為 Low (0)代表此時 TI 輸入訊號資料為 BWT 輸入字串；當 enable 為 High 時，mode 為 High (1)代表此時 TI 輸入訊號資料為 sub 短字串。
TI	I	8	BWTE 電路資料輸入匯流排，用以輸入組成字串的字元對應的 ASCII 碼，其代表意義由 mode 決定。
valid	O	1	BWTE 電路輸出控制訊號，當 valid 為 High 時代表 BWTE 電路的輸出資料 TO、SA、LF 為有效。可參考 2.4 系統時序部分的說明。
mode_o	O	2	BWTE 電路輸出控制訊號，當 mode_o 值為 2'b01 時代表當時 TO 上的資料為 BWT(T)；當 mode_o 值為 2'b10 時代表當時 TO 上的資料為進行 sub 短字串比對 hit 數目，佔據一個 clock cycle；當 mode_o 值為 2'b11 時代表當時 TO 上的資料為 sub 短字串比對定址出來的位置，佔用之 clock cycle 數視比對 hit 數決定。可參考 2.4 系統時序部分的說明。
TO	O	8	依 mode_o 的內容反映 BWTE 電路對 T 字串進行 BWT 轉換所得之 BWT(T)、或是進行 sub 短字串比對 hit 數目、或是 sub 短字串比對定址出來的位置，可參考 2.4 系統時序部分的說明。
SA	O	4	BWTE 電路對 T 字串進行 BWT 轉換所產生的 SA (Suffix Array)矩陣，可參考 2.4 系統時序部分的說明。
LF	O	4	BWTE 電路對 T 字串進行 BWT 轉換所產生的 LF_mapping 矩陣，可參考 2.4 系統時序部分的說明。
FINISH	O	1	告知 Host 端，BWTE 電路完成模擬，該訊號拉高 1 周期，代表這是最後一筆資料，請 Host 端開始進行評分。可參考 2.4 系統時序部分的說明。

## 2.3 系統描述

本 BWTE 電路設計的目的是對一長字串訊號進行 BWT 轉換，以及用一個短字串來做比對定位，參賽者應遵循題目指示，找出該長字串所對應的 BWT 轉換字串，並找出短字串在長字串中的比對結果，最後依題目指定的方式輸出題目所要求的結果。

### 2.3.1 BWT 輸入字串

雖然 BWT 在基因定位時所處理的字串由 A、C、G、T 四個字元組成，在此為方便起見，只考慮由 a、b、n 三個字元所組成長度為 6 字元的字串來示範 BWT 轉換。顯然的，用來做比對定位的 3 字元長度的短字串 sub 也應該由 a、b、n 這三個字元組成。

為方便起見，今以長度為 6 的字串「banana」來介紹 BWT 演算法及如何利用其結果對短字串「ana」來進行字串比對定位。

首先，為了進行 BWT 先將字串「banana」尾部加上「\$」標記，以此做為 **BWT 的輸入字串 T**，該字串 T 的字串長度為 7。

T = banana\$

strlen(T) = 7

上述之 strlen() 為 C 函數，可回傳字串長度。

## 2.3.2 BWT 演算法

由 T 字串建立字元矩陣 A，大小為 T 字串長度的方陣(此處即為 7x7 字元方陣)，在此描述 A 矩陣的列欄位置均由 0 開始算起，且以橫為列(row)，以縱為行(column)。A 的第 0 列(row)即為 T 字串本身；第 1 列為 T 字串向左 rotate 1 個單位(即 anana\$b)；第 2 列為 T 字串向左 rotation 2 個單位(即 anana\$b)；以此類推，A 的最後一列(第 6 列)為 T 字串向左 rotate 6 個單位(即\$banana)。

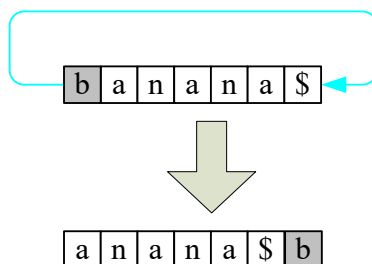


圖 4、向左 rotate 1 個單位操作示意圖

A =

b	a	n	a	n	a	\$	← 1st row
a	n	a	n	a	\$	b	← 2nd row
n	a	n	a	\$	b	a	
a	n	a	\$	b	a	n	
n	a	\$	b	a	n	a	
a	\$	b	a	n	a	n	
\$	b	a	n	a	n	a	← 7th row

↑ 1st column (F)  
 ↑ 7th column (L)

對字元矩陣 A 每一列(row)字串依字元由小到大進行排序(sorting)，令排序後的結果為字元矩陣 B (此即 Suffix Matrix，也稱為 BWT Matrix)。字元排序是依各字元的 ASCII 值來比大小，其中選作為標記的特殊字元\$必須是其中最小的。以本例來看，b 的 ASCII 值為 0x62 (0x 表示為 16 進位制表示法的值)、a 的 ASCII 值為 0x61、n 的 ASCII 值為 0x6e、且\$的 ASCII 值為 0x24。可將各列字串視為 256 進位字的數字表示法來進行大小比較，以進行排序。舉幾例說明下

字元 a (ASCII 為 0x61) 小於字元 n (ASCII 為 0x6e)

字串 ab (0x61-0x62) 小於字串 an (0x61-0x6e)

字串 \$a (0x24-0x61) 小於字串 an (0x61-0x6e)

字串 a\$n (0x61-0x24-0x6e) 小於字串 aab (0x61-0x61-0x62)

字元矩陣 A 作為排序後的結果字元矩陣 B 如下

B =

\$	b	a	n	a	n	a
a	\$	b	a	n	a	n
a	n	a	\$	b	a	n
a	n	a	n	a	\$	b
b	a	n	a	n	a	\$
n	a	\$	b	a	n	a
n	a	n	a	\$	b	a

則 T 字串的 BWT 轉換結果 BWT(T) 即為字元矩陣 B 每列最後一個字串所形成的字元陣列 (array)，此即 BWT matrix 的最後一行，又稱為 L 字元陣列(L array)。

B =

\$	b	a	n	a	n	a
a	\$	b	a	n	a	n
a	n	a	\$	b	a	n
a	n	a	n	a	\$	b
b	a	n	a	n	a	\$
n	a	\$	b	a	n	a
n	a	n	a	\$	b	a

L = annb\$aa  
F = \$aaabnn

$$\begin{aligned}
 L &= [B[6,0] \ B[6,1] \ B[6,2] \ B[6,3] \ B[6,4] \ B[6,5] \ B[6,6]] \\
 &= \text{annb\$aa} \\
 &= \text{BWT(T)}
 \end{aligned}$$

同理，由字元矩陣 B 每列第一個字串所形成的字元陣列，此即 BWT matrix 的第一行，稱為 F 字元陣列(F array)

$$\begin{aligned}
 F &= [B[0,0] \ B[0,1] \ B[0,2] \ B[0,3] \ B[0,4] \ B[0,5] \ B[0,6]] \\
 &= \$aaabnn
 \end{aligned}$$

因字元矩陣 B 是字元矩陣 A 經過排序而來，所以字元矩陣 B 中的每一列均存在字元矩陣 A 中，只是其列位置不同。描述該排序結果所代表的列位置調整關係稱為 Suffix\_array(以下用 SA 簡稱)，如字元矩陣 B 的第 0 列字串「\$banana」與字元矩陣 A 的第 6 列字串相同，即表示 B 矩陣的第 0 列字串是 A 矩陣第 6 列字串在排序過程中移動上來的結果，故 SA 的第 0 個元素即為 6。

同理，B 矩陣的第 1 列字串「a\$banan」為 A 矩陣的第 5 列，故 SA[1]=5。B 矩陣的第 2 列字串「ana\$ban」為 A 矩陣的第 3 列，故 SA[2]=3。...以此類推，得

$$SA = [6 \ 5 \ 3 \ 1 \ 0 \ 4 \ 2]$$

SA 的各元素也可解釋其對應的 BWT matrix 列字串是由 T 字串進行幾次左旋操作得來的。

由此建立 L array 中每個元素對應在 F array 的位置，即為 LF\_mapping 陣列(以下用 LF 簡稱)。  
 例如 L[0]元素 a 為 L array 中的第 1 個 a，與 A array 中的第 1 個 a(位置為 1)對應(為同一個)，所以 LF[0]=1；  
 L[1]元素 n 為 L array 中的第 1 個 n，與 A array 中的第 1 個 n(位置為 5)對應，所以 LF[1]=5；  
 L[2]元素 n 為 L array 中的第 2 個 n，與 A array 中的第 2 個 n(位置為 6)對應，所以 LF[2]=6；  
 ...以此類推，得

$$LF = [1 \ 5 \ 6 \ 4 \ 0 \ 2 \ 3]$$

利用 F 字串陣列、L 字串陣列(即 BWT 字串)、及 LF mapping 陣列搭配 BWT 的性質可推得 T 字串(即 Inverse-BWT 轉換)。

### 2.3.3 利用 BWT 做短字串比對定位

現就「banana」字串中進行短字串「ana」(以 sub 做代號)的比對定位(ExactMatching)。

Property 1: 由於每一列的組成字串都是 T 字串通過一連串的左旋產生的，所以每一個字串的最右邊一個字元，在原来的 T 字串中，會是同一字串的最左邊一個字元的左邊。

對 T 字串每一個字元依所在位置加上下標編碼以便觀察，加上下標的 T 字串如下：

$$T = \text{banana}\$ \Rightarrow \mathbf{b_0} \mathbf{a_1} \mathbf{n_2} \mathbf{a_3} \mathbf{n_4} \mathbf{a_5} \mathbf{\$6}$$

T 經過左旋兩次後的字串如下

$$\text{ROT-LEFT}(T, 2) = \mathbf{n_2} \mathbf{a_3} \mathbf{n_4} \mathbf{a_5} \mathbf{\$6} \mathbf{b_0} \mathbf{a_1}$$

該左旋兩字的字串為字元矩陣 B 的最後一列，該列的最左邊字元為  $n_2$ ，最右邊字元為  $a_1$ ，這兩個字元在原先的 T 字串( $b_0 \mathbf{a_1} \mathbf{n_2} a_3 n_4 a_5 \$6$ )中位置，可觀察到最右邊字元  $n_2$  在原 T 字串中位置在最左邊字元  $a_1$  的左邊(前面、字串頭方向)。

同理，字元矩陣 B 中的每一列均是由 T 字串經過不同程度的左旋構成，故有一樣的性質，故每個 L 字元矩陣的元素均是對應的 F 字元矩陣元素在 T 字串排列的前一個字元。

Property 2: 由於 BWT matrix 是經過字元大小排序的結果，所以 F 字元陣列本身是依大小排序過的，且由於 BWT matrix 的每一列是 T 字串經過一次次左旋操作所形成，所以 F 字元陣列的元素與 T 字串的元素組成是一樣的，比如 T 字串有幾個「a」字元，F 字元陣列就有幾個「a」字元。所以單獨算 F 字元陣列的話，也可以直接對 T 字串依字元 ASCII 值進行排序得到。



Property 3: 由於 BWT matrix 是經過字元大小排序的結果，所以 BWT matrix 中每一列從「\$」往前看到第一行位置，對應到 T 字串各種可能長度的字尾(Suffix)字串(參考下圖)。所以若是存在一短字串是 T 字串的子字串，該短字串必是 BWT matrix 其中一個 Suffix 字串的子字串。

\$	B	A	N	A	N	a
A	\$	B	A	N	A	N
A	N	A	\$	B	A	N
A	N	A	N	A	\$	B
B	A	N	A	N	A	\$
N	A	\$	B	A	N	A
N	A	N	A	\$	B	A

所以 F 字元陣列本身是依大小排序過的，且由於 BWT matrix 的每一列是 T 字串經過一次次左旋操作所形成，所以 F 字元陣列的元素與 T 字串的元素組成是一樣的，比如 T 字串有幾個「a」字元，F 字元陣列就有幾個「a」字元。所以單獨算 F 字元陣列的話，也可以直接對 T 字串依字元 ASCII 值進行排序得到。

所以當要在 T 字串中尋找 ana 短字串的存在位置時，應從最後一個字元(此處是「a」字元)開始著手，再利用 Property 1 的關係，一步步逆推檢查。首先觀察 BWT matrix 中代表 F 陣列的第一行元素滿足是最後一個字元(「a」字元)的部分(如圖 xx 第一行中第 1 到第 3 列綠色部分所示)

\$	B	A	N	A	N	a
A	\$	B	A	N	A	N
A	N	A	\$	B	A	N
A	N	A	N	A	\$	B
B	A	N	A	N	A	\$
N	A	\$	B	A	N	A
N	A	N	A	\$	B	A

由此知道 T 字串的子字串中滿足字串尾是字元「a」的有 3 個可能位置(對應到圖 xx 第一行中第 1 到第 3 列綠色部分所示，即 F[1]、F[2]、F[3])，利用 Property 1，這 3 個可能解的倒數第 2 個字元為各自對應的 L 矩陣字元元素(即 L[1]、L[2]、L[3])。從圖 xx 來看，可知現階段 3 個可能解的最後 2 個字元分別是「na」、「na」、及「ba」，其中只有前兩個是滿足 sub 字串特徵(倒數第 2 個字元為「n」)如圖 xx 中黃色所對應部分(即 L[1]、L[2])，剩兩個可能解。利用 LF\_mapping 換算這兩個 n 字元在 F 字元矩陣為 F[5]、F[6]。

此時可由 BWT matrix 的第 5、6 兩列看到 T 字串包含 na 短字串(如圖 xx1 中綠色部分)，這兩個「na」再往前拼接一個字元，其前一個字元(sub 字串的倒數第 3 個字元)為各自對應的 L 矩陣字元元素(即 L[5]、L[6])。由此可知現階段可能解有 2 個位置，且其字串均符合「ana」的特徵。因比對的 sub 短字串長度為 3，所以比對至此知道 T 字串中有 2 組「ana」子字串，此即比對的 **hit** 數為 2。

\$	B	A	N	A	N	a
A	\$	B	A	N	A	N
A	N	A	\$	B	A	N
A	N	A	N	A	\$	B
B	A	N	A	N	A	\$
N	A	\$	B	A	N	A
N	A	N	A	\$	B	A

將方才的 L[5]、L[6]再透過 LF-mapping 找到其在 F 字元陣列的位置分別為 2、3(即 LF[5]、LF[6])。此時可透過 Suffix Array 得到 ana 短字串在 T 字串的位置為 SA[2]及 SA[3]，即第 3 個位置與第 1 個位置。(記的再 check position index 是由 0 開始還是從 1 開始，盡量統一)

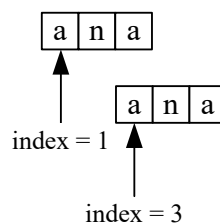
\$	B	A	N	A	N	a
A	\$	B	A	N	A	N
A	N	A	\$	B	A	N
A	N	A	N	A	\$	B
B	A	N	A	N	A	\$
N	A	\$	B	A	N	A
N	A	N	A	\$	B	A

index = 

0	1	2	3	4	5	6
---	---	---	---	---	---	---

T = 

b	a	n	a	n	a	\$
---	---	---	---	---	---	----



### 2.3.4 圖形相似度量測(Similarity)

現在說明物件與標的圖的比對方法。為說明方便，用一小尺寸的例子做說明。假設存在一尺寸為  $3 \times 4$  的影像  $\text{Img1}(x,y)$  及一尺寸為  $2 \times 3$  的影像  $\text{Img2}(x,y)$ ；如圖 12 所示，兩者均以其左上角為原點，水平為  $x$  軸，向右為正，垂直為  $y$  軸，向下為正。

圖 5、說明例之  $\text{Img1}(x,y)$  及  $\text{Img2}(x,y)$

這是一個捲積(convolution)運算的過程，寫成演算法可用一個 pseudo code 來表示如下：

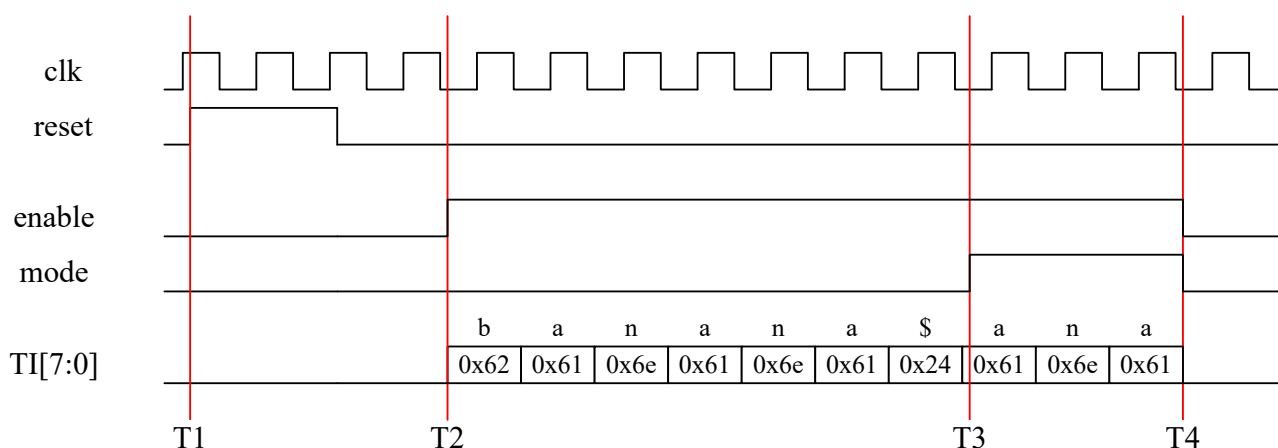
ExactMatching 的演算法要給嗎?

上面演算法及圖 13 中的  $\text{no2}()$  函數代表 number of 2 的縮寫， $\text{no2}(\text{Img1}(x, y), \text{Img2}(x, y))$  為計算同時存在  $\text{Img1}(x,y)$  跟  $\text{Img2}(x,y)$  兩張圖上面的 pixel 數目大小。圖 13 中的  $\text{no1}()$  函數代表 number of 1 的縮寫， $\text{no1}(\text{Img1}(x, y), \text{Img2}(x, y))$  為計算單獨存在  $\text{Img1}(x,y)$  或  $\text{Img2}(x,y)$  其中一張圖上面的 pixel 數目大小。 $\text{no2}() - \text{no1}()$  即代表  $\text{Img1}(x,y)$  及  $\text{Img2}(x,y)$  兩圖間的相似度，值越高代表兩圖間越相似。以另一個方式來說，將  $\text{Img1}(x,y)$  跟  $\text{Img2}(x,y)$  相加，相加後值為 2 的 pixel 數目即為該兩圖的相似度指標。(是否可用別的方式？比如改用點對點乘法，然後計算值是 1(此時值是 1 代表該座標在  $\text{Img1}$  及  $\text{Img2}$  均有值)的個數？或是直接對整個矩陣加起來？或是用 AND、OR、NOT、XOR 之類的運算?)

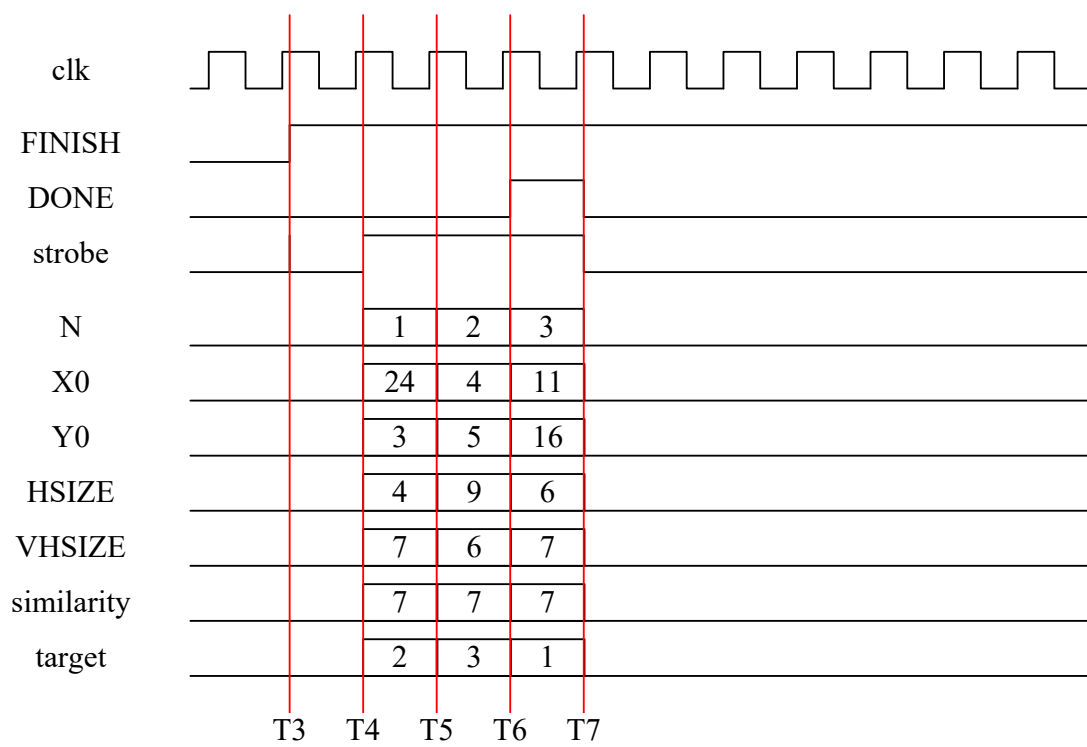
## 2.4 BWTE 電路輸出訊號及時序規格

BWTE 輸入輸出對應的時序波形圖如下(參考圖 15)

1. T1 ~ T2 時間點，BWTE 電路初始化。
2. 自 T2 時間點起，可依需要自行決定對 ROM 及 SRAM 進行讀寫動作。
3. 當找出整張圖所有前景物件的相似標的對象後，即可對 HOST 送出 FINISH 高準位訊號(即 T3 時間點)，代表計算完成，開始進行評分。採時脈正緣動作。
4. 於 T4 時間點開始每個前景物件圖像的相似度搜尋結果，此 cycle 內 strobe 訊號需拉為 High，代表**此周期內**的輸出資料有效，每個前景物件的搜尋結果輸出一個 cycle，進一步說明參考下一項。
5. 接著於 T4~T6 時間點依照前景物件的編號順序，輸出各自的「前景物件編號 N」、「前景物件原點 x 座標 X0」、「前景物件原點 y 座標 Y0」、「前景物件寬度 HSIZE」、「前景物件高度 VSIZE」、「該前景物件的最大相似度指標 similarity」、「及其相似標的物代碼 target」。標的物 t1 的代碼為 1，標的物 t2 的代碼為 2，標的物 t3 的代碼為 3。這幾個輸出對應的 strobe 訊號需拉為 High 代表**此周期內**的輸出資料有效。
6. 評分階段的輸出可一個一個個別輸出，但要用 strobe 訊號標示出該周期是有效的輸出(如圖 16)，**每筆輸出時間長度均為 1 周期**。
7. 於最後一筆資料輸出的時間周期內，需將 DONE 訊號**拉為 High 一個時脈周期**。Testbench 依此訊號來判斷輸出是否完成。**若 DONE 未拉為 High 則不予計分**。
8. 在此以 T7~T1 的時間代表所設計電路運算所需的總 cycle 數。



(a)



(b)

圖 6、(a)電路時序圖(前半段)、(b)電路評分輸出部分之時序圖(輸出訊號採連續輸出)

### 3. 評分標準

評分方式會依設計完成程度，分成 A、B、C、D 四種等級，排名順序為 A>B>C>D。由下面的評分項目來決定成績等級

- R1: RTL simulation 中，對其 BWT(T)、SA、LF 均回答正確，輸出有滿足時序波形圖之規定
- R2: RTL simulation 中，對其 N、POS 均回答正確，輸出有滿足時序波形圖之規定
- R3: Gate-level simulation 中，對其 BWT(T)、SA、LF、N、POS 均回答正確，輸出有滿足時序波形圖之規定
- R4: 在滿足 R1 要求的情形下，其設計之電路可合成(synthesizable)
- R5: 請注意**波形圖中最後一筆輸出 FINISH 訊號必須拉高 1 周期，然後要清為 0，若未滿足這項要求者不予計分，直接歸於「等級 D」**
- R6: 利用主辦單位所提供的 testbench 輸出訊息之 T7 (參考圖 17)，令三次測試輸入所得的三個 T7 相加後乘以該電路之面積(area, 單位  $\mu\text{m}^2$ , 參考圖 18)  
$$(T7\_pattern1 + T7\_pattern2 + T7\_pattern3) * area$$
其中 area 為 report\_area 的輸出 total cell area 一項

註：在某些狀況下，改用三次測試輸入所得的三個 T7 的和做排序依據(請參考後續說明)

```
VSIM 19> run -all
# Image Similarity Search Engine Start ...
# System is waiting for your finish signal ...
# Round 1
#   Output:  N= 1, (X0,Y0)=( 24, 3) (HSIZE,VSIZE)=( 4, 7), simi= 7, target= 2, DONE=0
#   Expect:   1           ( 24, 3)           ( 4, 7)           7           2           0
# Round 2
#   Output:  N= 2, (X0,Y0)=( 4, 5) (HSIZE,VSIZE)=( 9, 6), simi= 7, target= 3, DONE=0
#   Expect:   2           ( 4, 5)           ( 9, 6)           7           3           0
# Round 3
#   Output:  N= 3, (X0,Y0)=( 11, 16) (HSIZE,VSIZE)=( 6, 7), simi= 7, target= 1, DONE=1
#   Expect:   3           ( 11, 16)          ( 6, 7)           7           1           1
#
# Result
# Check all N, X0, Y0, HSIZE, VSIZE :    0 error
# Check all similarity, target       :    0 error
# T7 = 55 cycles
# ** Note: $finish : /user/DSD/htwu/w/c/vsim/archie/icc2021/tb.v(370)
# Time: 702 ns Iteration: 0 Instance: /tb
# 1
```

圖 7、testbench 輸出訊息中之 T7 輸出(示意圖，數值僅供參考)

Combinational area:	66509.224784
Buf/Inv area:	5767.765275
Noncombinational area:	17617.314754
Macro/Black Box area:	0.000000
Net Interconnect area:	undefined (No wire load specified)
<b>Total cell area:</b>	<b>84126.539538</b>
Total area:	undefined

1

圖 8、report\_area 輸出訊息中之 total cell area(示意圖，數值僅供參考)

設計完成度分四種等級，如下：

◇ **等級 A：**

- RTL 模擬結果完全正確。(即滿足 R1、R2 兩項目)
- 與 Gate-Level 模擬結果完全正確。(即滿足 R3、R4 兩項目)
- 此等級之成績計算方式如下：

$$\text{Score} = (\text{T7\_pattern1} + \text{T7\_pattern2} + \text{T7\_pattern3}) * \text{area}$$

註：Area 為合成軟體 Area report 得到的 Total cell area(um2) (參考圖 18)

註：本等級中，**Score 越小者**為同級名次越好!

◇ **等級 B：**

- RTL 模擬結果完全正確。(即滿足 R1、R2 兩項目)
- 無法 Synthesis，或 Gate-Level 模擬結果出現錯誤。
- 此等級之成績計算方式如下：

$$\text{Score} = (\text{T7\_pattern1} + \text{T7\_pattern2} + \text{T7\_pattern3})$$

註：本等級中，**Score 越小者**為同級名次越好!

◇ **等級 C：**

- RTL 模擬結果，滿足 R1 項目。
- RTL 模擬結果，未滿足 R2 項目。
- 此等級之成績計算方式如下：

$$\text{Score} = (\text{T7\_pattern1} + \text{T7\_pattern2} + \text{T7\_pattern3})$$

註：本等級中，**Score 越小者**為同級名次越好!

◇ **等級 D：**

- 未達成前三等級者或 FINISH 訊號未滿足時序圖之規定(即未滿足 R6 要求)。

## 附錄

附錄 A 為主辦單位所提供各參賽者的設計檔案說明

附錄 B 為主辦單位提供的測試樣本說明

附錄 B.2 為 Labeling 說明

附錄 C 為評分用檔案，亦即參賽者必須繳交的檔案資料

附錄 D 則為設計檔案壓縮整理步驟說明

附錄 E 中說明本次競賽使用之設計資料庫

附錄 F 中說明本次競賽之軟體環境



## 附錄 A、設計檔

1. 下表為主辦單位所提供各參賽者的設計檔

表 2、設計檔案說明

檔名	說明
BWTE.v	本題之設計檔 template，已包含系統 Input/Output Port 之宣告，請以此檔案作為 BWTE 電路之設計。
tb.v	本題僅有一個 testbench，內含有 tb、rom 及 sram 三個 module，有提供 3 組 test pattern 供驗證，因此請在模擬期間，自行使用+define+P1、+define+P2、+define+P3 參數，作 3 種 pattern 模擬時的切換。
pattern1.dat ~ pattern3.dat	ISSE 電路模擬時的輸入 pattern (即測試輸入影像)，共有 3 組。 註：這些檔案已加入至 tb.v，無需額外設定。
golden1.dat~golden3.dat	作為 ISSE 電路模擬時對應 3 種 pattern 的相似搜尋結果。 註：這些檔案已加入至 tb.v，無需額外設定。
BWTE_DC.sdc	Design Compiler 做電路合成之 Constraint 檔案，請自行設定 period 的期望值，但請勿更動環境相關參數。
dc_syn.tcl	dc 合成參考指令
.synopsys_dc.setup	Design Compiler 設定初始化範例檔案。參賽者請依 Library 實際擺放位置，自行修改 Search Path 設定
tsmc13_neg.v	Gate-level simulation 所需之 cell library file

2. 本題所提供之 testbench 檔，有增加數行 define 描述如下

```
`define SDFFILE " ./BWTE_syn.sdf "  
`ifdef SDF  
    initial  $sdf_annotate(`SDFFILE , u_ISSE);  
`endif
```

2.1 SDF 檔之檔名，請自行依 SDF 實際檔名及路徑進行修改後再模擬。

2.2 模擬。參賽者在進行 gate-level 模擬時，需於模擬命令上多加一個 +define+SDF 參數方可順利模擬，範例如下

```
ncverilog tb.v BWTE_syn.v -v tsmc13_neg.v +define+SDF
```

3. 主辦單位提供三組測試樣本供參賽者驗證設計的正確性，請自行使用+define+P1、+define+P2、+define+P3 參數作切換。
4. 請勿針對這三組測試樣本的內容湊答案作設計，評分時會另有隱藏的 pattern 作測試，若發現有針對測試樣本設計者，一律不予評分。
5. RTL 及 gate-level 模擬的指令下法以 ncverilog 說明如下

使用 P1 測試樣本之 RTL simulation 之 ncverilog 參考指令下法如下

```
ncverilog tb.v BWTE.v +define+P1
```

Gate-level simulation 參考指令下法如下

```
ncverilog tb.v BWTE_syn.v -v tsmc13_neg.v +define+P1+SDF
```

若要輸出 FSDB 檔，可自行在參數中加入 +define+FSDB +access+r

6. 以 dc\_shell 進行 synthesis 參考指令下法說明如下

```
source /cad/synopsys/CIC2/synthesis.csh
```

```
dc_shell -f dc_syn.tcl
```

## 附錄 B、測試樣本

本題提供 3 組測試樣本資料，如下：

第一組測試樣本資料 P1

## 第二組測試輸入影像 P2

Pattern2

第三組測試輸入影像 P3

	0										1										2										3									
	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1								
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
2	0	0	0	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
3	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
4	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1	0	0	0	0	0								
5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0								
6	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0								
7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0								
8	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
12	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
13	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
14	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
15	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0								
16	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0								
17	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	1	0	0	0	0	0								
18	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0								
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0								
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
24	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
25	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
26	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
27	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
28	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								

Pattern3	Obj1	Obj2	Obj3	Obj4	Obj5
X0	3	18	4	18	9
Y0	2	4	12	15	24
HSize	6	7	4	7	7
VSize	7	4	7	5	5
Similarity	1	7	7	7	5
Target	1 (t1 = T)	2 (t2 = F)	2 (t2 = F)	3 (t3 = 7)	1 (t1 = T)
N	1	2	3	4	5

### Pattern3

Obj1	t1 (T)	t2 (F)	t3 (7)
物件 0 度	<b>1</b>	-9	-9
物件 90 度	-17	-15	-18
物件 180 度	-11	-15	-12
物件 270 度	-20	-18	-18

Obj2	t1 (T)	t2 (F)	t3 (7)
物件 0 度	-4	-17	-14
物件 90 度	-1	<b>7</b>	-8
物件 180 度	-13	-17	-17
物件 270 度	-10	-14	-11

Obj3	t1 (T)	t2 (F)	t3 (7)
物件 0 度	2	<b>7</b>	-8
物件 90 度	-16	-14	-14
物件 180 度	-7	-14	-11
物件 270 度	-1	-11	-8

Obj4	t1 (T)	t2 (F)	t3 (7)
物件 0 度	-16	-17	-17
物件 90 度	-13	-14	-17
物件 180 度	-16	-20	-20
物件 270 度	-1	-8	<b>7</b>

Obj5	t1 (T)	t2 (F)	t3 (7)
物件 0 度	-16	-14	-17
物件 90 度	-7	-14	-8
物件 180 度	-19	-14	-20
物件 270 度	<b>5</b>	-8	<b>-2</b>

## 附錄 C、評分用檔案

評分所須檔案可以下幾個部份：(1)RTL design，即各參賽隊伍對該次競賽設計的 RTL code，若設計採模組化而有多個電路模組(module)，請務必將合成所要用到的各 module 檔放到同一個 ISSE.v 內，且 ISSE 模組須放在第一個，以免評審進行評分時，無法進行模擬，違者將酌情調整成績；(2)Gate-Level design，即由合成軟體所產生的 gate-level netlist，以及對應的 SDF 檔。

表 3

<b><i>RTL category</i></b>		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
N/A	N/A	Design Report Form (繳交紙本)
RTL Simulation	ISSE.v (or ISSE.sv)	Verilog 或 System Verilog synthesizable RTL code
<b><i>Gate-Level category</i></b>		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
Pre-layout Gate-level Simulation	ISSE_syn.v	Verilog gate-level netlist generated by Synopsys Design Compiler
	ISSE_syn.sdf	SDF timing information generated by Synopsys Design Compiler
	ISSE_syn.ddc	Design database generated by Synopsys Design Compiler

## 附錄 D、檔案整理步驟

當所有的文件準備齊全如表 3 所列，請按照以下的步驟指令，提交相關設計檔案，將所有檔案複製至同一個資料夾下，步驟如下：

1. 在自己的 home directory 建立一個新目錄，名稱叫做“**result**” 例如：  
    > **mkdir ~/result**
2. 將附錄 D 要求的檔案複製到 result 這個目錄。例如：  
    > **cp BWTE.v ~/result/**  
    > **cp BWTE\_syn.v ~/result/**  
    其他檔案依此類推.....
3. 若所設計電路包含多個模組，請將所有電路模組放到同一個 BWTE.v (或 BWTE.sv)內，且 ISSE 模組須放在第一個。
4. 在 Design Report Form 中，填入所需的相關資訊。

## 附錄 E、設計資料庫

設計資料庫位置： **/usr/cad/icc2021/CBDK\_IC\_Contest\_v2.5**

目錄架構

SynopsysDC/		
db/		
	slow.db	Synthesis model (slow)
lib/		
	slow.lib	timing and power model
Verilog/		
	tsmc13_neg.v	Verilog simulation model



## 附錄 F、軟體環境

1. 使用者登入後自動會設定好以下軟體環境：

Vendor	Tool	Executable
Cadence	Virtuoso *1	icfb
	Composer	icfb
	NC-Verilog	ncverilog
	Innovus	innovus
Synopsys	Design Compiler	dv, dc_shell
	VCS-MX	vcs
	IC Compiler	icc_shell -gui
	Hspice	hspice
	Cosmos Scope *1	cscope
	Custom Explorer *1	wv
	Laker *1	laker
	Laker ADP*1	adp
	Verdi *1	verdi, nWave, nLint
Mentor	Calibre *3	calibre
	QuestaSim, ModelSim	vsim
Utility	vi	vi, vim
	gedit	gedit
	nedit	nedit
	acoread	acoread
	calculator	gnome-calculator, bc -l
	gcc	Gcc
	Matlab	matlab

EDA 軟體所須使用的 license 皆已設定完成，不須額外設定

\*1 該軟體限定使用 1 套 license

\*3 該軟體限定使用 3 套 license

## Design Report Form

<b>登入帳號(login-id)</b>		
<b><i>RTL category</i></b>		
<b><i>Design Stage</i></b>	<b><i>Description</i></b>	<b><i>File Name</i></b>
RTL Simulation	使用之 HDL 名稱 (例如：Verilog、System Verilog)	
	RTL 檔案名稱 (RTL file name, BWTE.v or BWTE.sv)	
	RTL simulation, 所使用的 CYCLE 數 (請 確定模擬功能正確) T7 (cycles)	P1 ( ) cycles P2 ( ) cycles P3 ( ) cycles
<b><i>Gate-Level category</i></b>		
<b><i>Design Stage</i></b>	<b><i>Description</i></b>	<b><i>File Name</i></b>
Pre-layout Gate-level Simulation	Gate-Level 檔案名稱 (*_syn.v)	
	Pre-layout Gate-Level sdf 檔案名稱 (*_syn.sdf)	
	Design database 檔案名稱 (*_syn.ddc)	
	Gate-Level simulation, 所使用的 CYCLE 數 (請確定模擬功能正確) T7 (cycles)	P1 ( ) cycles P2 ( ) cycles P3 ( ) cycles
	Total cell area (um <sup>2</sup> ) (以小數點後做四捨五入)	( ) um <sup>2</sup>
<b><i>Grade</i></b>		
	最後完成之等級?(ex: 等級 A)	
其他說明事項(Any other information you want to specify:(如設計特點 ...)如寫不下可寫於背面		