

2014 University/College IC Design Contest

Cell-Based IC Design Category for Graduate Level

String Matching Engine

1.問題描述

請完成一 String Matching Engine (**後文以 SME 表示**)的電路設計，其功能為，將整篇英文文章的字串(**後文以 String 表示**)依照題目指定的關鍵字樣本(**後文以 Pattern 表示**)，搜尋該文章所有字元，凡是在 String 中有符合題目指定的 Pattern 者，將比對成功的字串(**後文以 Matched String 表示**)第一個字元的記憶體位址值作輸出，並輸出比對成功的 Pattern 所屬之編號值。

整篇文章以 ASCII 碼呈現並放置在 rom_4096x8_t13 記憶體裡，題目所指定的 Pattern 則放置於 rom_128x8_t13 記憶體裡，在搜尋 String 過程中，有滿足本題指定 Pattern 之一者，隨時可將其 Matched String 的位址作輸出(亦即**本系統可接受亂序輸出**)，當整篇文章搜尋完畢後，須將 finish 訊號拉為 High，以告知系統搜尋、輸出皆完畢。有關 SME 詳細的運作方式，將描述於後。

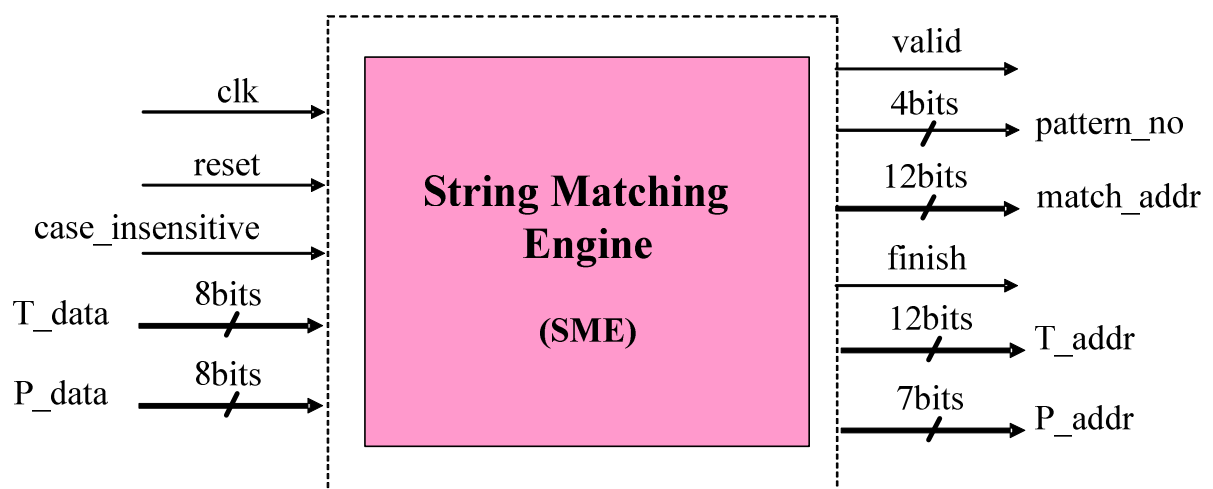
本電路各輸入輸出信號的功能說明，請參考表 1。每個參賽隊伍必須根據下一節所給的設計規格及附錄 A 中的測試樣本完成設計驗證。

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，CIC 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 E 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/icc2014/bgcc/icc2014cb.tar**，請執行以下指令取得測試樣本：

```
tar xvf /usr/cad/icc2014/bgcc/icc2014cb.tar
```

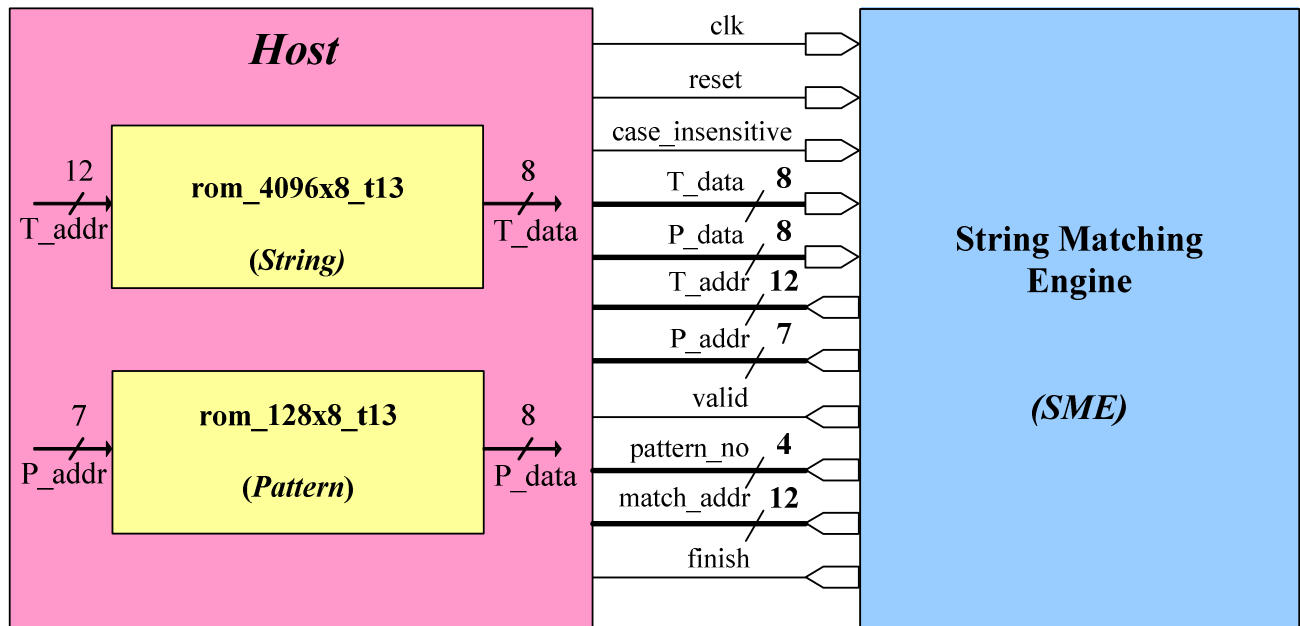
軟體環境及設計資料庫說明請參考附錄 F 與附錄 G。



圖一、String Matching Engine 之 I/O 腳位圖

2.設計規格

2.1 系統方塊圖



圖二、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	本系統為同步於時脈正緣之同步設計。
reset	I	1	高位準”非”同步(active high asynchronous)之系統重置信號。
case_insensitive	I	1	指定大、小寫英文字母視為相同與否。當此訊號為 0，表示大、小寫的字母視為不同(例如：大寫 A 與小寫 a，視為不同)；反之當此訊號為 1，表示大、小寫的字母視為相同(例如：大寫 A 與小寫 a，視為相同)。
T_addr	O	12	String ROM 的位址匯流排。英文文章的字串(String)已擺置於 rom_4096x8_t13 內，當要讀取 String 時，需先送 T_addr 的位址給 rom_4096x8_t13，該 ROM 便會將該記憶體位址內存放的 ASCII 碼，藉由 T_data 讀入 SME 電路。。
T_data	I	8	String ROM 的資料匯流排。根據 T_addr 所指定的位址，將該位址內存放的 ASCII 碼，透過 T_data 讀入 SME 電路。

P_addr	O	7	Pattern ROM 的位址匯流排。欲搜尋關鍵字的樣本 (Pattern)已擺置於 rom_128x8_t13 內，當要讀取 Pattern 時，需先送 P_addr 的位址給 rom_128x8_t13，該 ROM 便會將該記憶體位址內存放的 ASCII 碼，藉由 P_data 讀入 SME 電路。
P_data	I	8	Pattern ROM 的資料匯流排。根據 P_addr 所指定的位址，將該位址內存放的 ASCII 碼，透過 P_data 讀入 SME 電路。
pattern_no	O	4	Matched String 是成功比對到哪一組 Pattern，將 Matched Pattern 編號值，透過此匯流排輸出。 該輸出訊號必須與 match_addr 同一個 Cycle 內一起輸出。
match_addr	O	12	Matched String 的第一個位址，透過此匯流排輸出。 該輸出訊號必須與 pattern_no 同一個 Cycle 內一起輸出。
valid	O	16	有效的輸出訊號。當 valid 為 High，表示目前輸出的 pattern_no、match_addr 資料為有效的輸出，反之，當 valid 為 Low，表示 pattern_no、match_addr 為無效的輸出。
finish	O	1	SME 運算完成之通知訊號。未完成搜尋前，該訊號請保持為 Low。當所有 Matched String 的 pattern_no、match_addr 都輸出完成後，需將 finish 訊號拉為 High，以告知 Host 端搜尋、輸出皆完畢。 若系統一直未收到 finish 訊號拉為 High，模擬將永無止境。

表 2 – ASCII 碼字元表

ASCII 碼			ASCII 碼			ASCII 碼			ASCII 碼		
十進位	十六進位	字元	十進位	十六進位	字元	十進位	十六進位	字元	十進位	十六進位	字元
032	20		056	38	8	080	50	P	104	68	h
033	21	!	057	39	9	081	51	Q	105	69	i
034	22	"	058	3A	:	082	52	R	106	6A	j
035	23	#	059	3B	;	083	53	S	107	6B	k
036	24	\$	060	3C	<	084	54	T	108	6C	l
037	25	%	061	3D	=	085	55	U	109	6D	m
038	26	&	062	3E	>	086	56	V	110	6E	n
039	27	'	063	3F	?	087	57	W	111	6F	o
040	28	(064	40	@	088	58	X	112	70	p
041	29)	065	41	A	089	59	Y	113	71	q
042	2A	*	066	42	B	090	5A	Z	114	72	r
043	2B	+	067	43	C	091	5B	[115	73	s
044	2C	,	068	44	D	092	5C	\	116	74	t
045	2D	-	069	45	E	093	5D]	117	75	u
046	2E	.	070	46	F	094	5E	^	118	76	v
047	2F	/	071	47	G	095	5F	_	119	77	w
048	30	0	072	48	H	096	60	`	120	78	x
049	31	1	073	49	I	097	61	a	121	79	y
050	32	2	074	4A	J	098	62	b	122	7A	z
051	33	3	075	4B	K	099	63	c	123	7B	{
052	34	4	076	4C	L	100	64	d	124	7C	
053	35	5	077	4D	M	101	65	e	125	7D	}
054	36	6	078	4E	N	102	66	f	126	7E	~
055	37	7	079	4F	O	103	67	g			

註：換行符號，ASCII 碼十進位為 10，十六進位為 0A

2.3 系統描述

SME 電路功能為，將已知的英文文章字串(String)中，搜尋關鍵字樣本(Pattern)，將該 String 內有 Matched String 的第一個位址予以輸出，並輸出其 Pattern 編號。在此以舉例方式說明 SME 電路細項運作如下：

2.3.1 String 來源與存放方式

假設一篇英文文章為：

原始文章:

AGCCDTA GCc
Tccdtaa GTC

依照表 2 將該文章轉換成 ASCII 碼，並存放在 rom_4096x8_t13 記憶體裡，如下表所示：

T	ROM: rom_4096x8_t13																									
	String	A	G	C	C	D	T	A		G	C	c	換行	T	c	c	d	t	a	a		G	T	C	換行	文章結束
	Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
	Data (ASCII)	41	47	43	43	44	54	41	20	47	43	63	0A	54	63	63	64	74	61	61	20	47	54	43	0A	0

註：本範例中的 ASCII 碼，係以十六進位表示。

註：文章換行處，其 ASCII 碼的十六進位為 0A。

註：文章結束處，會以 00 作為結束的識別點。

2.3.2 Pattern 符號說明與存放方式

在本範例中，假設給定七組 Pattern 為：

P	Pattern0	C	C	D	T	A
	Pattern1	A	?	G	C	
	Pattern2	G	C			
	Pattern3	G	.	C		
	Pattern4	^	T			
	Pattern5	C	\$			
	Pattern6	A	.	G	?	.

註：此七組 Pattern 純作舉例之用，並非競賽時所使用的 Pattern。

註：非黃色 Hilight 處，表示該字元不存在。

其中

特殊符號說明	
?	前一字元比對零次或一次
.	比對任意字元
^	比對字串開頭
\$	比對字串結尾

註：本題只會用到此四組特殊符號。

註：本題的特殊符號“？”最多只會連續出現 2 個，但同一組 Pattern 特殊符號”？“最多 4 個。

註：本題的特殊符號”。“不會連續出現，同一組 Pattern 特殊符號”。“最多 2 個。

因此將本範例之七組 Pattern 存放於 rom_128x8_t13，儲存方式如下：

P	ROM: rom_128x8_t13																										
	Address	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17		
	Data (ASCII)	43	43	44	54	41	00	41	3F	47	43	00	47	43	00	47	2E	43	00	5E	54	00	43	24	00		
	註解	Pattern0						Pattern1						Pattern2				Pattern3				Pattern4			Pattern5		
	ROM: rom_128x8_t13																										
	Address	18	19	1A	1B	1C	1D	1E																			
	Data (ASCII)	41	2E	47	3F	2E	00	00																			
	註解	Pattern6					樣本結束																				

註：本範例中的 ASCII 碼，係以十六進位表示。

註：每組 Pattern 皆會有一組 00 隔開儲存，Pattern 結束處則會有兩組 00 的訊號。

2.3.3 SME 的輸出

SME 電路隨 case_insensitive 訊號不同，會有兩種輸出結果如下表：

Output of SME (for this example)	case_insensitive = 0															
	pattern_no:	0	1	1	2	2	3	3	4	5	6	6				
	match_addr:	2	0	8	1	8	1	14	C	16	0	6				
	case_insensitive = 1															
	pattern_no:	0	0	1	1	2	2	3	3	3	4	5	5	6	6	6
	match_addr:	2	D	0	8	1	8	1	8	14	C	A	16	0	6	11

註：此表輸出順序，僅供參考，真實 SME 輸出順序，只要 pattern_no 與 match_addr 兩訊號在同一 Cycle 內搭配一起輸出，本題即可接受亂序輸出。

說明如下：

1. 當 case_insensitive 為 0：

a. pattern_no=0、match_addr=2

⇒ 滿足 CCDTA 只有 rom address：02 ~ 06，因只需輸出第一個位址，故輸出此值。

b. pattern_no=1、match_addr=0

⇒ 字元 A，比對到一次，故輸出為此值。而 Pattern1 的 A?GC，因為該樣本已輸出 pattern_no=1、match_addr=0，故針對 Pattern1 最後兩個字元的 GC，不用再輸出 pattern_no=1、match_addr=1 這組值，請注意！

c. pattern_no=1、match_addr=8

⇒ 字元 A，比對到零次，故輸出此值。

d. pattern_no=3、match_addr=1

⇒ G 與 C 中間確實有比對到任意字元，故輸出此值。

e. pattern_no=4、match_addr=C

⇒ 只有在 String ROM 位址 C 為文章段落的開頭處，同時比對成功到 T 字元，故輸出此值。

f. pattern_no=5、match_addr=16

⇒ 在 String ROM 位址 A、16，皆有字母 C，但考量大小寫視為不同，故僅輸出此值。

2. 當 case_insensitive 為 1：

a. pattern_no=0、match_addr=D

⇒ 因大小寫英文字母視為相同，故此值可輸出。

註：上述範例，僅供 SME 細部運作說明用，此簡易範例中的 T 與 P，非競賽時使用，請注意！

當 Matched String 搜尋到，請將 valid 訊號拉為 High，並在同一個 Cycle 內，將 pattern_no 與 match_addr 的配對組合作輸出即可。當 Host 端收到 valid 訊號為 High，便會開始進行資料比對，比對結果有 pass 或 error 兩種狀況，**pass 表示本次輸出組合正確無誤，確實存在於正確解答中，error 表示本次輸出組合，並未存在於正確解答中。**當所有 Matched String 皆輸出完成，請將 finish 訊號拉為 High，以告知 Host 端所有 Matched String 皆已輸出完畢，系統便會開始展開 miss 的資料比對，所謂 **miss 意指：當實際輸出筆數與正確解答筆數不同，則表示有部分 Matched String 未被搜尋到，將被一一列出其 miss 的 pattern_no 與 match_addr**，表示 SME 電路功能上仍有問題，請注意！

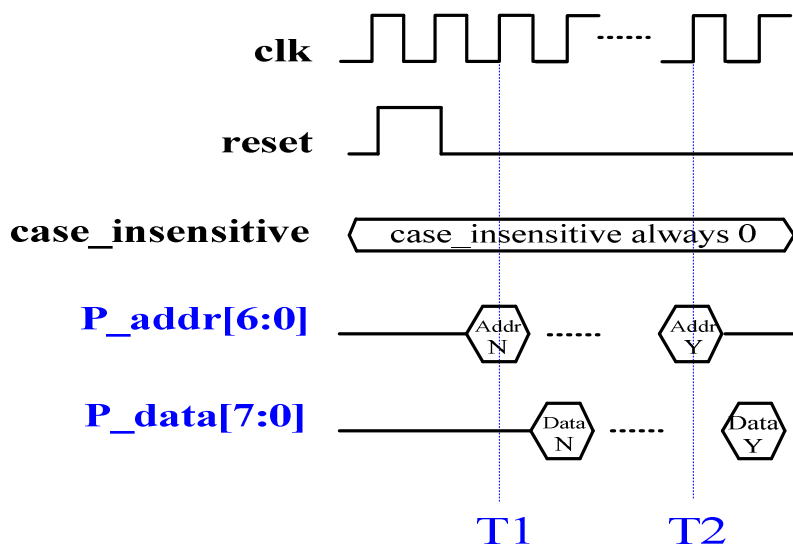
註：本系統之 valid 訊號拉為 High，系統會檢驗輸出的資料為 pass 或 error。

註：本系統之 finish 訊號拉為 High，系統會檢驗輸出的資料有無完整，有無 miss。

2.4 SME 電路時序規格

2.4.1 載入 Pattern 之時序規格

本時序圖，只先針對 Pattern 載入作說明，SME 比對電路的時序圖請參考 2.4.2。本時序圖只針對 case_insensitive 為 0 作舉例，同理 case_insensitive 為 1 其時序圖也依此類推。在整個模擬過程中，case_insensitive 會一直維持在 0 或 1，表示本次模擬，SME 電路是否會區別英文字母大、小寫。



1. T1 時間點，reset 一個 Cycle 的時間，SME 電路初始化結束，參賽者隨即可開始送出位址訊號，經過 ROM 的 t_a 存取時間後，Pattern 的資料即可被讀取。注意：ROM 的資料讀取，本題並"未"規定一定要從位址 0 開始讀取，請參賽者依照實際需求，自行安排位址之讀取順序。

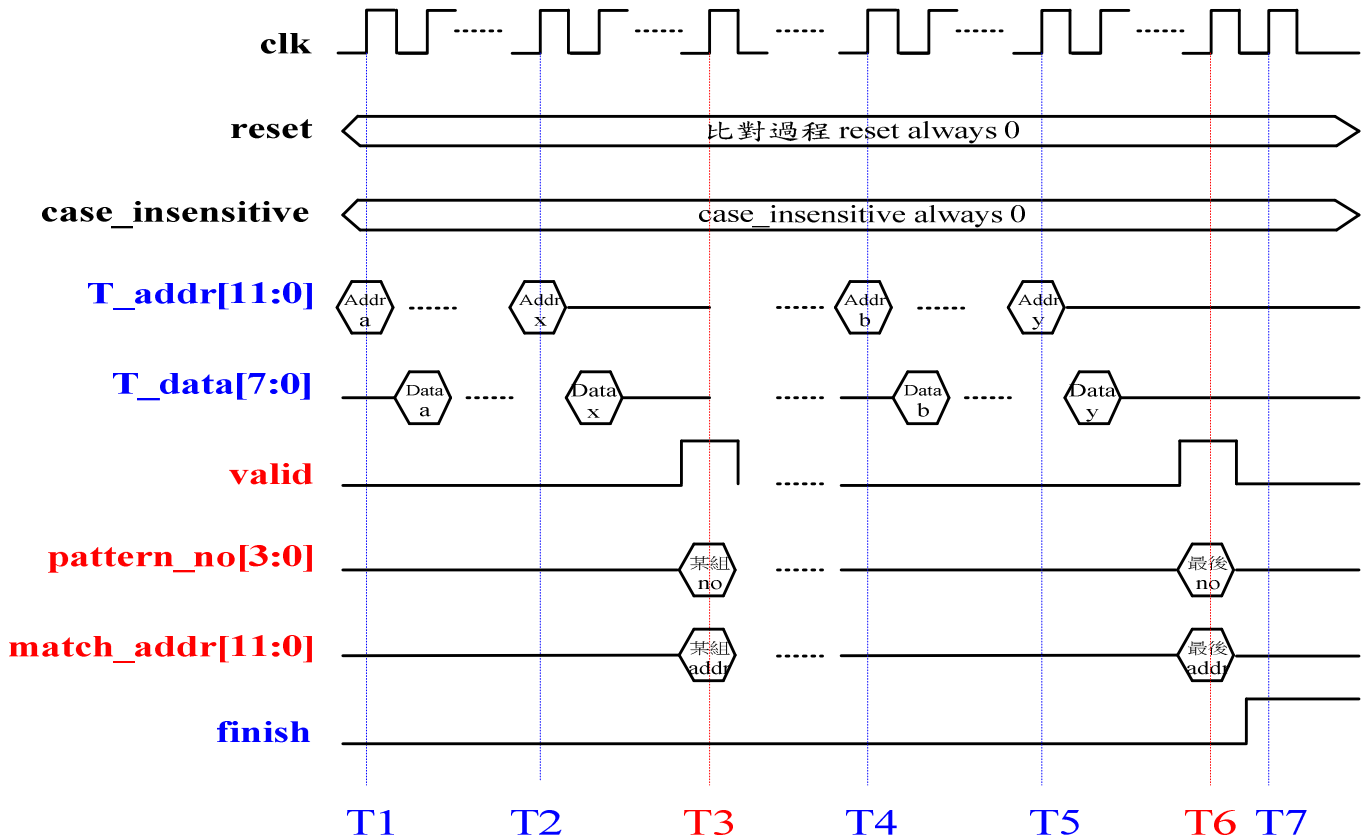
註：有關 ROM 的 t_a 存取時間與 Read-Cycle 時序圖，詳細內容請參考 ROM 的 PDF 檔。

註：本題並"未"規定所有 Pattern 必須要先 Load 完，才可以開始作比對，請依照實際需求作讀取順序控制。

2. T2 時間點，假設 Addr Y 為 Pattern 的最後一個字元之位址值，讀取完成後，Pattern 便全數載入完畢。

2.4.2 載入 String 及輸出 Matched String 之時序規格

本時序圖為 SME 電路載入 String 資料與 Matched String 作輸出之時序說明圖。本時序圖只針對 case_insensitive 為 0 作舉例，同理 case_insensitive 為 1 其時序圖也依此類推。在整個模擬過程中，case_insensitive 會一直維持在 0 或 1，表示本次模擬，SME 電路是否會區別英文字母大、小寫。



1. 當 Pattern 載入適當數量的字元或全數載入時，String 的資料可依自己的 schedule 進行載入，T1 時間點，假設為此次 schedule 欲載入的第一個位址值，T2 時間點，假設為此次 schedule 欲載入的最後一筆位址值，分別經過 ROM 的 t_a 存取時間後，該 String 的字元即可被讀取，並進行資料比對。
2. T3 時間點，當 Matched String 搜尋到，將 valid 訊號拉為 High，並在同一個 cycle 內送出 Matched String 的 pattern_no 與 match_addr 值。Host 端收到 valid 為 High 即可立即判斷出，該值為 pass 或 error。
3. 接下來的搜尋與 Load String 方式與上述兩條說明相同。
4. T4~T5 時間點，假設為最後一次 schedule 的第一個位址值與最後一個位址值，分別經過 ROM 的 t_a 存取時間後，該 String 的字元即可被讀取，並進行資料比對。
5. T6 時間點，最後一組 Matched String 被搜尋到，將 valid 訊號拉為 High，並在同一個 cycle 內送出 Matched String 的 pattern_no 與 match_addr 值。

6. T7 時間點，當最後一組 Matched String 都輸出後，請將 finish 訊號拉為 High，以告知 Host 端所有 pattern_no 與 match_addr 皆以輸出，系統便會開始進行 miss 資料比對，比對完成後，整個模擬立即結束。

註：miss 的資料比對期間，不會浪費任何 Simulation Time。

2.4.2 ROM 記憶體規格與時序規格

本題 Host 端有兩顆記憶體，分別儲存 String 資料於 rom_4096x8_t13 與 Pattern 資料於 rom_128x8_t13。有關此兩種 ROM 細項規格與時序圖，詳如記憶體 PDF 檔案中。

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，評分項目有兩個，分別為**模擬時間**、**面積**，主辦單位會依此兩項目做為同等級之評分。另外，**請參賽者提供一組正確的週期時間(CYCLE TIME)**給評分人員驗證本電路之正確性。

✧ 評分項目一：依”模擬時間”(Time)長短評分

各參賽隊伍將 APR 完成後，執行 Gate-level Post-layout Simulation 模擬完後，會出現模擬時間，評分人員會以此模擬時間如下面範例，紀錄成 **Time =10190.572 ns** 做評分。

```
***** The Summary of String matching *****
-----
pass   :          247
error  :           0
miss   :           0
-----
-----
Congratulations! All data have been generated successfully!
-----PASS-----
Simulation complete via $finish(1) at time 10190572 PS + 0
./testfixture1.v:210      #(`CYCLE/2); $finish;
```

註：Simulation Time 若不同則以最大值為準。

✧ 評分項目二：依”面積”(Area)大小評分

各參賽隊伍將 APR 完成後，面積分析方法如下範例，請任選其一 APR 軟體做分析。

1. IC Compiler Report Area 範例:

```
icc_shell> get_attribute [get_die_area] bbox
{0.000 0.000} {316.360 316.270}
=> Area = 316.36x316.27 = 100055.1772 um2
```

2. Encounter Report Area 範例:

encounter > **analyzeFloorplan**

```
***** Analyze Floorplan *****
Die Area(um^2)           : 100328.96
Core Area(um^2)          : 93974.36
Number of instance(s)    : 8662
Number of Macro(s)       : 0
Number of IO Pin(s)      : 56
Number of Power Domain(s) : 0
***** Estimation Results *****
*****
=> Area = 100328.96 um2
```

註：指令 **analyzeFloorplan 會破壞已完成 routing 的結果，執行該指令後絕對不可再存檔。**

設計完成程度四種等級，如下：

☆ **等級 A：達成”完成設計”之三項要求**

- a、功能正確，RTL 模擬與正確解答比對完全正確。
- b、完成 Synthesis，且 Gate-Level Pre-layout Simulation 結果正確。
- c、**完成 APR，並達成 APR 必要項目**，Gate-Level Post-layout Simulation 結果正確。

註：完成 APR 之必要項目

- i. 只需做 Marco layout (即不用包含 IO Pad、Bonding Pad)。
- ii. VDD 與 VSS Power Ring 寬度請各設定為 **2um**，**只須做一組**。
- iii. **不需加 Dummy Metal**。
- iv. **Power Stripe 務必至少加一組**，其 VDD、VSS 寬度各設定為 **2um**。
(Power Stripe 垂直方向至少一組，水平方向可不加)
- v. **務必要加 Power Rail (follow pin)**。
- vi. Core Filler 務必要加。
- vii. APR 後之 GDSII 檔案務必產生。
- viii. 完成 APR，DRC/LVS **完全無誤(見附錄 C 說明)**。

等級 A 之評分方法：

當 $\text{Area} < 40000\mu\text{m}^2$ $\text{Score} = \text{Area} \times \text{Time}^2$

當 $\text{Area} > 40000\mu\text{m}^2$ $\text{Score} = 2 \times \text{Area} \times \text{Time}^2$

例如：

本範例(Encounter)為，

$$\text{Score} = 2 \times \text{Area} \times \text{Time}^2 = 2 \times 100328.96 \times 10190.572^2 = 20837875054174.35209728$$

註：Score 越小者，同級名次越好！

◇ **等級 B：已做到 APR，但等級 A 之”APR 必要項目”有部分不符合，DRC/LVS 錯誤總數量容許 5 個(含)以下**

此等級之成績計算方式如下：

當 $\text{Area} < 40000\mu\text{m}^2$ $\text{Score} = \text{Area} \times \text{Time}^2 \times (\text{DRC} + \text{LVS 的錯誤總數量})$

當 $\text{Area} > 40000\mu\text{m}^2$ $\text{Score} = 2 \times \text{Area} \times \text{Time}^2 \times (\text{DRC} + \text{LVS 的錯誤總數量})$

註：Score 越小者，同級名次越好！

◇ **等級 C：僅完成合成，或做到 APR，但 DRC/LVS 錯誤總數量超過 5 個以上**

此等級之成績計算方式如下：

當 $\text{Area} < 40000\mu\text{m}^2$ $\text{Score} = \text{Area} \times \text{Time}^2$

當 $\text{Area} > 40000\mu\text{m}^2$ $\text{Score} = 2 \times \text{Area} \times \text{Time}^2$

註：

1. Score 越小者，同級名次越好！
2. 等級 C，視 APR 為 Fail，Area 以 Design Compiler 所 Report 的 Cell Area 為主。
3. 等級 C，視 APR 為 Fail，Time 以 Gate-level Pre-layout Simulation 為主。

◇ **等級 D：未達成前三等級者，成績計算方式為 RTL Simulation，比對結果之 miss 與 error 總數量越少者，分數越高。**

$$\text{Score} = \text{Total (miss + error) of All RTL Simulations}$$

註：

1. 等級 D，Score 評分方式為所有模擬的 error 及 miss 總數作相加。
2. 等級 D，視合成及 APR 皆為 Fail，Area、Timing 將不予考慮之。
3. 等級 D，只以 RTL Simulation 正確率為主，Score 越小者(即 error & miss 越少)，同級名次越好。

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 C 為設計驗證說明；附錄 D 為評分用檔案，亦即參賽者必須繳交的檔案資料；附錄 E 則為設計檔案壓縮整理步驟說明；附錄 F 中說明本次競賽之軟體環境；附錄 G 中說明本次競賽使用之設計資料庫。

附錄 A 設計檔(For Verilog)

1. 下表為主辦單位所提供各參賽者的設計檔

表 3、設計檔案說明

檔名	說明
SME.v	參賽者所使用的設計檔，已包含系統輸/出入埠之宣告。
testfixture1.v testfixture2.v	第一組 Pattern 之 Test Bench 檔案 (其內容詳如附錄 B)。此兩個 Test Bench 主要差異是 case_insensitive 分別設定為 0 與 1，並已設定完成正確解答之檔案。
testfixture3.v	第二組 Pattern 之 Test Bench 檔案 (其內容詳如附錄 B)。此 Test Bench case_insensitive 設定為 1，並已設定完成正確解答之檔案。
rom_4096x8_t13.v rom_4096x8_t13_verilog.rcf	String ROM 的模擬檔案與其 Data。由於 String ASCII 碼存放於 rom_4096x8_t13_verilog.rcf，模擬時請務必將該檔案放置同一個目錄裡。
rom_128x8_t13_pattern1.v rom_128x8_t13_pattern1.rcf	第一組 Pattern ROM 的模擬檔案與其 Data。由於第一組 Pattern ASCII 碼存放於 rom_128x8_t13_pattern1.rcf，模擬時請務必將該檔案放置同一個目錄裡。
rom_128x8_t13_pattern2.v rom_128x8_t13_pattern2.rcf	第二組 Pattern ROM 的模擬檔案與其 Data。由於第二組 Pattern ASCII 碼存放於 rom_128x8_t13_pattern2.rcf，模擬時請務必將該檔案放置同一個目錄裡。

string_ascii.dat	String ROM Data 細項註解說明檔。該 Data 內容與 rom_4096x8_t13_verilog.rcf 完全相同，唯一差別為本檔案提供 String 每個字元各別 ASCII 對照表與特殊符號註解說明，方便參賽者設計時純參考用。
pattern1_verilog.dat	Pattern 1 ROM Data 細項註解說明檔。該 Data 內容與 rom_128x8_t13_pattern1.rcf 完全相同，唯一差別為本檔案提供 Pattern 每個字元各別 ASCII 對照表與特殊符號註解說明，方便參賽者設計時純參考用。
pattern2_verilog.dat	Pattern 2 ROM Data 細項註解說明檔。該 Data 內容與 rom_128x8_t13_pattern2.rcf 完全相同，唯一差別為本檔案提供 Pattern 每個字元各別 ASCII 對照表與特殊符號註解說明，方便參賽者設計時純參考用。
golden1.dat golden2.dat golden3.dat	三種模擬結果之正確解答檔案。第一組 Pattern 當 case_insensitive 為 0，正確解答檔案為 golden0.dat。第一組 Pattern 當 case_insensitive 為 1，正確解答檔案為 golden2.dat。第二組 Pattern 當 case_insensitive 為 1，正確解答檔案為 golden3.dat。
.synopsys_dc.setup	使用 Design Compiler 作合成或 IC Compiler Layout 之初始化設定檔。參賽者請依 Library 實際擺放位置，自行修改 Search Path 的設定。注意：無論合成或 APR，只需使用 worst case library。
SME_DC.sdc	Design Compiler 作合成之 Constraint 檔案。參賽者可依需求自行修改 cycle 的設定，其餘環境參數請勿更改。
SME_APR.sdc	Encounter 或 IC Compiler 作 Layout 之 Constraint 檔案。參賽者可依需求自行修改 cycle 的設定，其餘環境參數請勿更改。

請使用 *SME.v*，進行 SME 電路之設計。其模組名稱、輸出/入埠宣告如下所示：

```
module SME ( clk, reset, case_insensitive, pattern_no, match_addr,
             valid, finish, T_data, T_addr, P_data, P_addr);

    input      clk;
    input      reset;
    input      case_insensitive;
    output [3:0] pattern_no;
    output [11:0] match_addr;
    output      finish;
    input [7:0] T_data;
    output [11:0] T_addr;
    input [7:0] P_data;
    output [6:0] P_addr;

endmodule
```

2. 本題有兩組 Pattern(其內容詳如附錄 B)又因 case_insensitive 有 0 或 1，因此在此提供三種 Test Bench：*testfixture1.v*、*testfixture2.v*、*testfixture3.v* 分別對應到第一組 Pattern case_insensitive=0、第一組 Pattern case_insensitive=1、第二組 Pattern case_insensitive=1，並將其相關會用到的檔案皆已加入，如下：

例如：

當作第一組 Pattern 且 case_insensitive=0：請使用 *testfixture1.v* 作模擬：

```
`define EXP          "/golden1.dat"
parameter E_NUM      = 247;
```

註：參賽者無須作修改，只需注意 *golden1.dat*、*rom_4096x8_t13_verilog.rcf*、*rom_128x8_t13_pattern1.rcf* 的檔案位置即可，這三個檔案的預設路徑為”目前目錄”。

3. 本題所提供的 Test Bench 檔案，有多增加幾行特別用途的敘述如下：

```
`define End_CYCLE    100000000
`define SDFFILE       "/SME_syn.sdf"
`ifdef SDF
    initial $sdf_annotate(`SDFFILE, SME);
`endif
```

註：

1. *End_CYCLE* 預設 10000 萬個 Cycles，其目的可以防止參賽者因電路有錯，模擬陷入無窮回圈之境，倘若參賽者確定模擬真的需要超過 10000 萬個 Cycles 以上，請自行加大此 Cycle 數。
2. SDF 檔案，請自行修改 SDF 實際檔名及路徑後再模擬。

3. 在 Test Bench 中，主辦單位提供`ifdef SDF 的描述，其目的是讓本 Test Bench 可以作為 RTL 模擬、合成後模擬與 Layout 後模擬皆可使用。注意：當參賽者在合成或 Layout 後模擬，請務必多加一個參數`+define+SDF`，方可順利模擬。

例如：當合成後，使用 NC-Verilog 模擬第一組樣本，在 UNIX 下執行下面指令

```
> ncverilog +ncmaxdelays testfixture1.v SME_syn.v rom_4096x8_t13.v
           rom_128x8_t13_pattern1.v -v tsmc13_neg.v +define+SDF +access+rw
```

例如：當合成後，使用 NC-Verilog 模擬第二組樣本，在 UNIX 下執行下面指令

```
> ncverilog +ncmaxdelays testfixture2.v SME_syn.v rom_4096x8_t13.v
           rom_128x8_t13_pattern1.v -v tsmc13_neg.v +define+SDF +access+rw
```

例如：當合成後，使用 NC-Verilog 模擬第三組樣本，在 UNIX 下執行下面指令

```
> ncverilog +ncmaxdelays testfixture3.v SME_syn.v rom_4096x8_t13.v
           rom_128x8_t13_pattern2.v -v tsmc13_neg.v +define+SDF +access+rw
```

註：使用 Encounter 作 APR 之參賽者，APR 後的模擬務必加上+ncmaxdelays 參數，請注意！

```
ncsim> run
FSDB Dumper for IUS, Release Verdi3_2013.07, Linux, 07/04/2013
(C) 1996 - 2013 by Synopsys, Inc.
*Verdi3* FSDB WARNING: The FSDB file already exists. Overwriting the FSDB file
using this file.
*Verdi3* : Create FSDB file 'SME.fsdb'
*Verdi3* : Begin traversing the scopes, layer (0).
*Verdi3* : End of traversing.
*Verdi3* : Begin traversing the MDAs, layer (0).
*Verdi3* : Enable +mda and +packedmda dumping.
*Verdi3* : End of traversing the MDAs.
*****      The Summary of String matching      *****

-----

pass   :          247

error  :           0

miss   :           0

-----

-----

Congratulations! All data have been generated successfully!

-----PASS-----

Simulation complete via $finish(1) at time 10190572 PS + 0
./testfixture1.v:210      #(`CYCLE/2); $finish;
```

附錄 B 測試樣本

本題所提供之長篇英文文章 String 局部內容如下：**(本題共計 4020 個字元)**

Little Redcap

The wolf lifted the latch the door sprang open and without saying a word he went straight to the grandmothers bed and devoured her Then he put on her clothes dressed himself in her cap laid himself in bed and drew the curtains

:

:

So now the wretch went for the third time to the housedoor knocked at it and said Open the door for me children your dear little mother has come home and has brought every one of you something back from the forest with her The little kids cried First show us your paws that we may know if you are our dear little mother

上述文章轉為 ASCII 碼，並附加註解，可參考 string_ascii.dat 檔案。在此秀出局部內容：

```
4C      // Char= L   Dec:076
69      // Char= i   Dec:105
74      // Char= t   Dec:116
74      // Char= t   Dec:116
6C      // Char= l   Dec:108
65      // Char= e   Dec:101
20      // Char=     Dec:032
52      // Char= R   Dec:082
65      // Char= e   Dec:101
64      // Char= d   Dec:100
63      // Char= c   Dec:099
61      // Char= a   Dec:097
70      // Char= p   Dec:112
0A      // Char= LF (NL line feed, new line)
0A      // Char= LF (NL line feed, new line)
54      // Char= T   Dec:084
68      // Char= h   Dec:104
65      // Char= e   Dec:101
20      // Char=     Dec:032
77      // Char= w   Dec:119
6F      // Char= o   Dec:111
6C      // Char= l   Dec:108
66      // Char= f   Dec:102
20      // Char=     Dec:032

:
```

註：資料左側為十六進位，右側為原始字元與十進位。

本題第一組 Pattern，內容如下：

P	Pattern0	L	i	t	t	l	e							
	Pattern1	g	r	a	n	d	m	o	t	h	e	r		
	Pattern2	h	i	m	s	e	l	f						
	Pattern3	T	h	e		b	e	t	t	e	r		t	o
	Pattern4	m	o	t	h	e	r							
	Pattern5	t	o											
	Pattern6	a	n	d										
	Pattern7	T	h	e	y	?								
	Pattern8	s	e	e	?									
	Pattern9	h	a	d	?									
	Pattern10	w	?	h	e	n								
	Pattern11	w	o	.										
	Pattern12	.		c	h	i	l	d						
	Pattern13	^	T											
	Pattern14	d	\$											

本題第二組 Pattern，內容如下：

P	Pattern0	o	.	g	?	h	?	t						
	Pattern1	s	o	m	?	e	?	.	h	?	i	?	.	

註 1：非黃色 Hilight 處，表示該字元不存在。

註 2：請注意 Pattern 每個字元大小寫與特殊符號，空白格其 ASCII 碼十六進位為 20。

第一組 Pattern，轉為 ASCII 碼，並附加註解，可參考 pattern1_verilog.dat 檔案。在此秀出局部內容：

```

4C      // Char= L   Dec:076
69      // Char= i   Dec:105
74      // Char= t   Dec:116
74      // Char= t   Dec:116
6C      // Char= l   Dec:108
65      // Char= e   Dec:101
00      // end of pattern
67      // Char= g   Dec:103
72      // Char= r   Dec:114
61      // Char= a   Dec:097
6E      // Char= n   Dec:110
64      // Char= d   Dec:100
6D      // Char= m   Dec:109
6F      // Char= o   Dec:111
74      // Char= t   Dec:116
68      // Char= h   Dec:104
65      // Char= e   Dec:101
72      // Char= r   Dec:114
00      // end of pattern

```

註：資料左側為十六進位，右側為原始字元與十進位。

第二組 Pattern，轉為 ASCII 碼，並附加註解，同樣地請自行參考 pattern2_verilog.dat 檔案。

第一組 Pattern 且 case_insensitive=0，輸出結果請參考 golden1.dat 檔案，在此秀出局部內容：

```
0_000      //Dec:0
0_0F4      //Dec:244
0_975      //Dec:2421
1_070      //Dec:112
1_183      //Dec:387
1_284      //Dec:644
1_30B      //Dec:779
1_35B      //Dec:859
1_3BE      //Dec:958
1_411      //Dec:1041
1_459      //Dec:1113
1_6F3      //Dec:1779
1_860      //Dec:2144
2_0B5      //Dec:181
2_0CD      //Dec:205
2_5AB      //Dec:1451
2_B24      //Dec:2852
2_DA3      //Dec:3491

:
```

註：資料左側為十六進位，在符號”_”左側為 pattern_no，符號”_”右側為 match_addr。

註：資料右側為左側數值之十進位數值。

同理，第一組 Pattern 且 case_insensitive=1，輸出結果請參考 golden2.dat 檔案，第二組 Pattern 且 case_insensitive=1，輸出結果請參考 golden3.dat 檔案，在此不再贅述。

附錄 C 設計驗證說明

參賽者繳交資料前應完成 RTL，Gate-Level 與 Physical 三種階段驗證，以確保設計正確性。**注意：**每組限定只能使用 1 license，勿使用 Multi-CPU。

- RTL 與 Gate-Level 階段：參賽者必須進行 RTL simulation 及 Gate-Level simulation，模擬結果必須於參賽者提供之 CYCLE 數值下，功能完全正確。
- Physical 階段，包含三項驗證重點：
 1. 依主辦單位各項要求，實現完整且正確的 layout (詳細之各項要求，請見評分標準)。
 2. 完成 post-layout simulation：參賽者必須使用 P&R 軟體**寫出之 netlist 檔與 sdf 檔完成 post-layout gate-level simulation**，以下分為 IC Compiler、Encounter 兩種軟體說明 netlist 與 sdf 寫出步驟。

- i. 使用 Synopsys IC Compiler 者，執行步驟如下：

在 IC Compiler 主視窗底下點選

“File > Export > Write SDF...”

Specify Version	Version 2.1
Instance	空白即可
File name	SME_pr.sdf
Significant digits	2

按 。

對應指令： write_sdf -version 2.1 SME_pr.sdf

“File > Export > Write Verilog...”

先按

Output verilog file name	SME_pr.v
Output physical only cells	disable
Wire declaration	enable
Backslash before Hierarchy Separator	Enable
All other options	Default value

按 。

- ii. 使用 Cadence Encounter 者，執行步驟如下：

在 Encounter 視窗下點選：

“File → Save → Netlist...”

Netlist File	SME_pr.v
All other options	Default value

按 。

“Timing → Extract RC...”，按 。

“Timing → Write SDF...”

Ideal Clock	Disable
SDF Output File:	SME_pr.sdf

按 **OK**。

3. **完成 DRC 與 LVS 驗證：參賽者必須以其所使用之 P&R 軟體內含之 DRC 與 LVS 驗證功能完成 DRC 與 LVS 驗證，以下分為 IC Compiler、Encounter 兩種軟體說明執行步驟。**

- i. 使用 Synopsys IC Compiler 者，驗證 DRC 與 LVS 步驟如下：

在 IC Compiler Layout 視窗底下點選

“Route > Verification > DRC ...”

Read child cell from	Cell view
All other options	Default value

按 **OK**。

將跳出 Error Browser 視窗，請參賽者自行查看是否有錯，若有請自行修改 Layout 到 0 個 Violation 為止。

“Route > Verification > LVS ...”

Pins not connected to a wire segment(Floating port)	disable
All other options	Default value

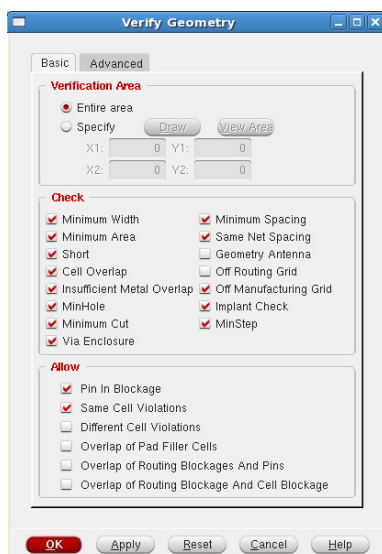
按 **OK**。

將跳出 Error Browser 視窗，檢查看看是否有錯，若有請自行修正到 0 個 Violation 為止。

- ii. 使用 Cadence Encounter 者，驗證 DRC 與 LVS 步驟如下：

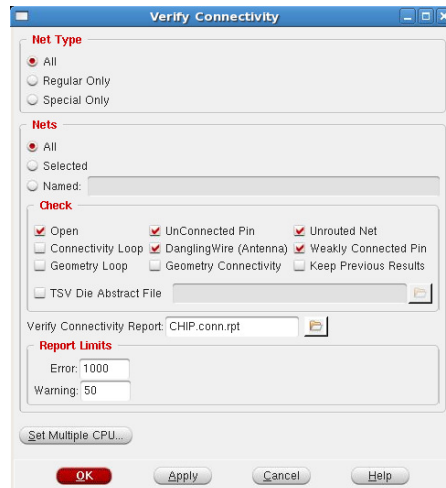
在 Encounter 視窗下點選

1. DRC 驗證：請選**“Verify → Verify Geometry...”** Default 值，按 **OK**。



註：若 DRC 有發生錯誤，請選**“Tools → Violation Browser...”**查明原因。

2. LVS 驗證：請選“Verify → Verify Connectivity...” Default 值，按 **OK**。



註：若 LVS 有發生錯誤，請選“Tools → Violation Browser...”查明原因。

附錄 D 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設計檔，請務必將合成所要用的各 module 檔放進來，以免評審進行評分時，無法進行模擬；(2)Gate-Level design，即由合成軟體所產生的 gate-level netlist，以及對應的 SDF 檔；(3)Physical design，使用 **Synopsys IC Compiler** 者，請記得將整個 **Milkyway Library** 等相關的 design database，壓縮成一個檔案。使用 **Cadence Encounter** 者，請將 Encounter 相關的 design database (包含 **.enc** 檔案與 **and .enc.dat** 目錄)，壓縮成一個檔案。壓縮的檔案格式如下：假設參賽者的 design database 目錄名稱爲”your_lib”，請執行底下的 UNIX 指令，最後可以得到”your_name.tar”的檔案。

```
> tar cvf your_name.tar your_lib
```

在執行以上的指令之前，請確定將你使用的 P&R Tool 儲存後關閉，再執行上述的指令，否則在壓縮的過程會出現錯誤。

表 4

RTL category		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
N/A	N/A	Design Report Form
RTL Simulation	*.v or *.vhd	Verilog (or VHDL) synthesizable RTL code
Gate-Level category		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
Pre-layout Gate-level Simulation	*_syn.v	Verilog gate-level netlist generated by Synopsys Design Compiler
	*_syn.sdf	Pre-layout gate-level sdf
Physical category		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
P&R	*.tar	archive of the design database directory
	*.gds	GDSII layout
	DRC/LVS report	不用儲存 DRC/LVS Report 檔案!只需在 Design Report Form 上填寫 DRC/LVS 錯誤總數量 即可。(目標要做到 0 個錯誤!)
Post-layout Gate-level Simulation	*_pr.v	Verilog gate-level netlist generated by Cadence Encounter or Synopsys IC Compiler
	*_pr.sdf	Post-layout gate-level sdf

附錄 E 檔案整理步驟

當所有的文件準備齊全如表 4 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 SME.v ~/result/
```

```
> cp SME_syn.v ~/result/
```

```
.....
```

3. 在 Design Report Form 中，填入所需的相關資訊。

附錄 F 軟體環境

1. 本題已將所有軟體環境設定於：/cad/icc2014/cshell/login.cshrc，參賽同學不需再做任何設定。

2. 在 env.cshrc 所設定好的軟體環境包括：

NC-Verilog

NC-VHDL

Verdi

Laker

VCS-MX

ModelSim

Design Compiler (Design Vision)

IC Compiler

Encounter

gedit / nedit

vi / vim

gvim

gnome-calculator (工程計算機執行檔，可開啟 View -> Scientific mode)

gcc

EDA 軟體所須使用的 license 皆已設定完成，不須額外設定，且每組限定**每個 EDA 軟體只能使用一套 license**。

附錄 G 設計資料庫

設計資料庫位置： /usr/cad/icc2014/CBDK_IC_Contest_v2.1

目錄架構

ICC/

tsmc13gfsg_fram/	ICC core library
tsmc13_CIC.tf	ICC technology
macro.map	layer mapping file
tluplus/	
t013s8mg_fsg_typical.tluplus	t13 tluplus file
t013s8mg_fsg.map	t13 tluplus mapping file

SOCE/

lef/	
tsmc13fsg_8lm_cic.lef	LEF for core cell
antenna_8.lef	LEF for antenna
lib/	
slow.lib	worst case for core cell
streamOut.map	Layout map for GDSII out

SynopsysDC/

db/	
slow.db	Synthesis model (slow)
lib/	
slow.lib	timing and power model

Verilog/

tsmc13_neg.v	Verilog simulation model
--------------	--------------------------

Phantom/

tsmc13gfsg_fram.gds	Standard Cell GDSII file
---------------------	--------------------------

Design Report Form

登入帳號(login-id)		
RTL category		
<i>Design Stage</i>	<i>Description</i>	<i>File Name</i>
RTL Simulation	使用之 HDL 名稱 (請填入 Verilog 或 VHDL)	
RTL Simulation	RTL 檔案名稱 (RTL Netlist file name)	
Gate-Level category		
<i>Design Stage</i>	<i>Description</i>	<i>File Name</i>
Pre-layout Gate-level Simulation	Gate-Level 檔案名稱 (Gate-Level Netlist file name)	
	Pre-layout sdf 檔案名稱	
	Gate-Level simulation, 所使用的 CYCLE Time (請確定模擬功能正確)	() ns
Physical category		
<i>Design Stage</i>	<i>Description</i>	<i>File Name or Value</i>
P&R	使用之 P&R Tool (請填入 IC compiler 或 Encounter)	
	設計資料庫檔案名稱 (Library name) (ICC: Milkyway Library Name, Encounter: xxx.enc.dat)	
	DRC 錯誤總數量 (ex: 0 個)	
	LVS 錯誤總數量 (ex: 0 個)	
	佈局檔檔案名稱 (GDSII file name)	
	佈局面積 (Layout Area)	() μm^2
Post-layout Gate-level Simulation	Gate-Level 檔案名稱 (Gate-Level Netlist file name)	
	Post-layout sdf 檔案名稱	
	Post-layout Simulation 所使用的 CYCLE Time (請確定模擬功能正確) (請寫出三組模擬中的最大值) Ex: 2ns	
	Post-layout Simulation Time (Simulation Time, ex: 10190 ns) Time = ? (請寫出三組模擬中的 最大值)	
Over All	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		