

2019 IC Design Contest

Cell-Based IC Design Category for Graduate Level

IoT Data Filtering

1.問題描述

請完成物聯網(IoT) Data Filtering(後文以 **IOTDF** 表示)的電路設計。本電路可將智慧裝置或感測器(Sensor)收集到的巨量 IoT 資料依據指定應用功能作即時分析處理，最後將處理結果後的數值以 `iot_out` 傳輸出，即為所求!由於考量到模擬時間問題，本題 IoT 資料量在此限定為固定 96 筆，每筆資料長度為 128bits，指定的應用功能(Function)共計七種。

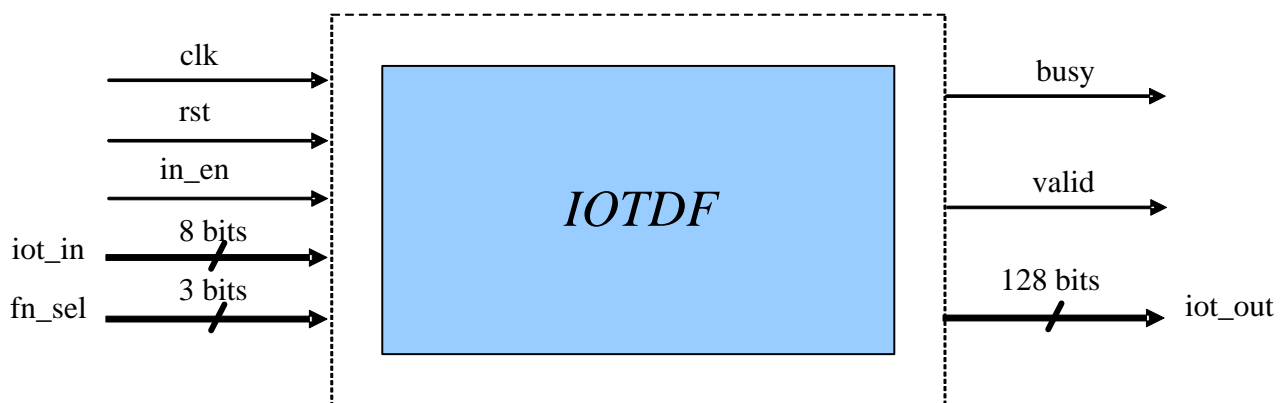
有關 IOTDF 詳細規格將描述於後。表一為本電路各輸入、輸出信號的功能說明。各參賽隊伍必須依照下一節所指定的設計規格及附錄 A 中的測試樣本完成設計驗證。

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，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 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/icc2019/bgc/icc2019cb.tar`**，請執行以下指令取得測試樣本：

```
tar xvf /usr/cad/icc2019/bgc/icc2019cb.tar
```

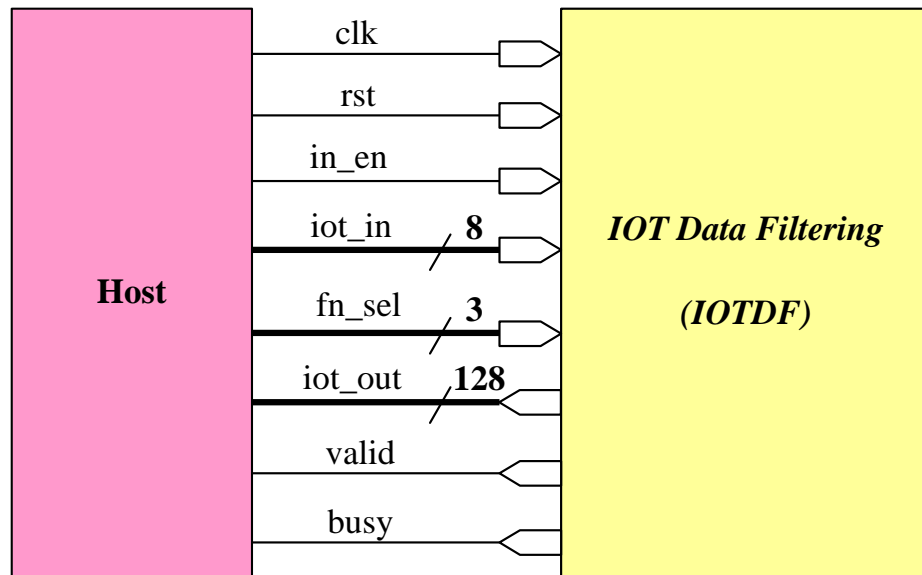
軟體環境及設計資料庫說明請參考附錄 F 與附錄 G。



圖一、IOT Data Filtering 之方塊圖

2.設計規格

2.1 系統方塊圖



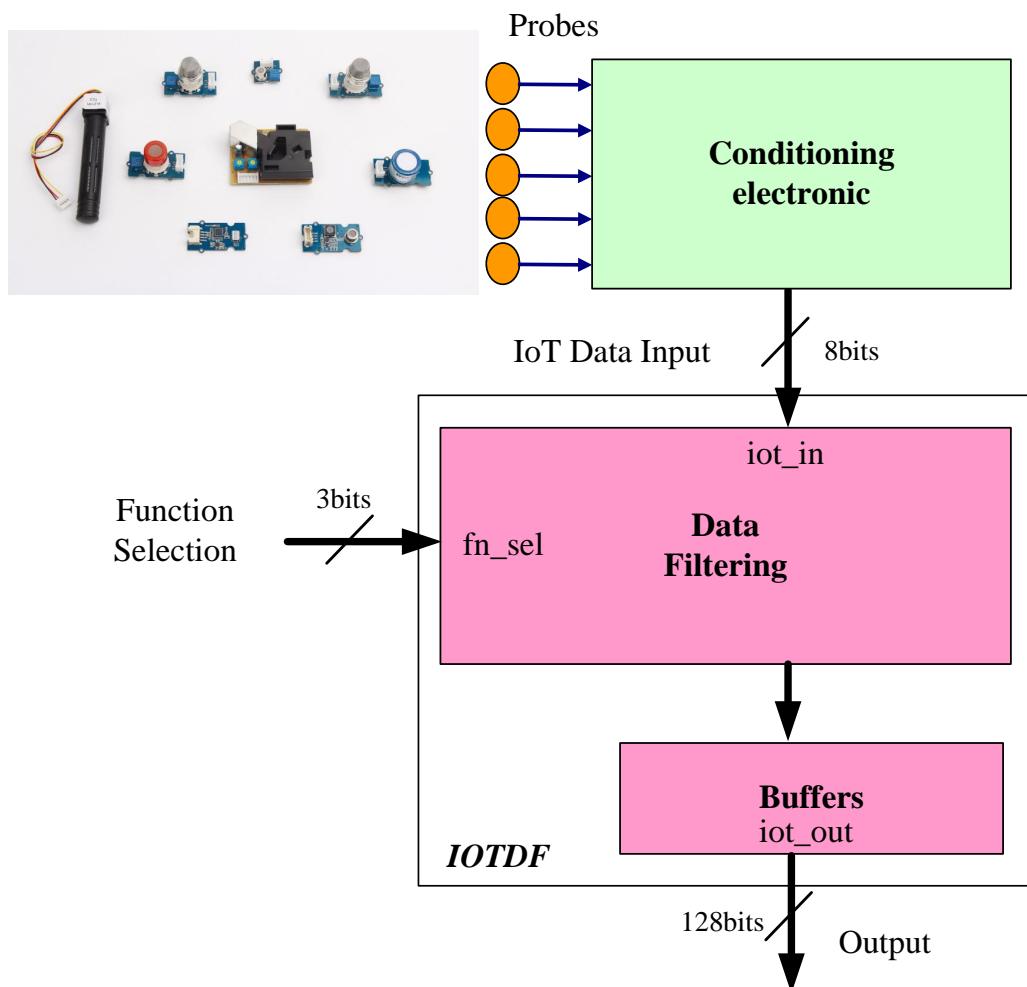
圖二、系統方塊圖

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	Clock Signal (positive edge trigger)
rst	I	1	Reset Signal (active high)。
in_en	I	1	Input Enable Signal。本訊號會與 busy 訊號作搭配，當 Host 端有資料要輸入時，此時 busy 訊號為 low，表示 IOTDF 電路可以接收新的一筆資料，in_en 因此為 high，反之，busy 訊號為 high 時，表示 IOTDF 電路尚未能接收新的一筆資料進來，in_en 因此為 low。當 Host 端所有資料輸入完畢後，該訊號到模擬結束前將永遠維持為 low。
iot_in	I	8	IOT Input Signal。Host 端會透過此匯流排將每筆 128bits 資料，每次以 8bits 作輸入，共計 16 個 Cycles 後即可送完一筆資料，接著再以相同方式輸入第二筆，直到 96 筆的資料輸入完畢為止。 註：詳細的輸入順序與方法，請參照 2.3.1。

fn_sel	I	3	Function Select Signal。本題有七種應用功能，透過此訊號可作為七種功能的切換，每次模擬只會固定選擇某一種功能，模擬過程中不會任意切換另一種功能。 <i>註：fn_sel 如何對應至七種應用功能，請參照 2.3.2。</i>
iot_out	O	128	IOT Output Signal。IOTDF 電路會透過此匯流排將運算完結果輸出至 Host 端，每次以 128bits 直接輸出(即輸出一筆資料僅需花 1 個 Cycles 時間)，直到被選定的應用功能其資料皆輸出後，模擬隨即結束。 <i>註：各種應用功能的輸出方法，請參照 2.3.3。</i>
busy	O	1	IOTDF Busy Signal。其動作已描述於 in_en 說明處。
valid	O	1	IOTDF Output Valid Signal。當為 high，表示目前輸出的 iot_out 資料為有效的輸出，反之則輸出為 low。



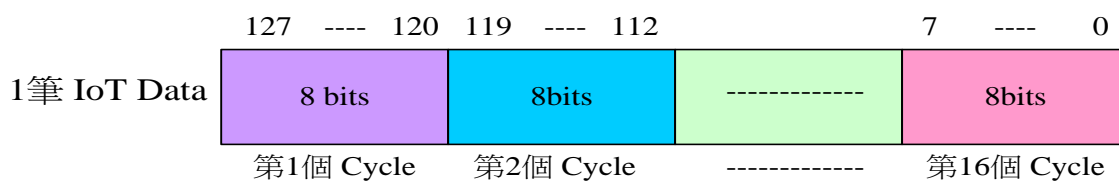
圖三、IoT Data Collection & IOTDF 電路架構

2.3 系統描述

物聯網(Internet of Things)的應用相當廣泛，例如智慧家庭應用，家中裝有多個溼度感測器，感測器收集到的數值傳送至 IOTDF 電路作分析，當溼度高於主人設定的 Max 值，便將此數據透過網路傳送至主人智慧手機的 APP 程式，告知主人現在家中的溼度過高了需要開啟除濕機囉，或是溼度低於主人設定的 Min 值，透過網路告知主人現在溼度是多少，已經相當低了，建議可以關閉除濕機了。如圖三所示，所有智慧裝置或感測器收集到的資料，傳輸至 IOTDF 電路，依據指定的應用功能作分析處理，例如當感測到的數值超過 Max 臨界值或是低於 Min 臨界值時，才將此數據傳送出去，以達到節能之功效，**本題僅需針對圖三中，粉色方塊的 IOTDF 電路作設計即可。**

2.3.1 IOTDF 電路的輸入

感測器收集到的資料傳送至 IOTDF 電路，無論是哪一種應用功能，**每一筆資料輸入**皆固定為**128bits 正整數**，分成**16 個 Cycles**傳送，採**MSB 的 8bits 先送**，傳送輸入方式如下圖：



註：無論是哪一種應用功能，**輸入資料的筆數**，皆固定為**96 筆**。

2.3.2 IOTDF 電路的七種應用功能

本電路之七個 Functions 是透過 fn_sel 訊號作選擇，其對照如下表：

表 2 - fn_sel 與七種應用功能之對應表

	fn_sel	Functions
F1	3'b001	Max(N)
F2	3'b010	Min(N)
F3	3'b011	Avg(N)
F4	3'b100	Extract(low < data < high)
F5	3'b101	Exclude(data<low , high<data)
F6	3'b110	PeakMax(該 data 比之前已輸出的數值都大)
F7	3'b111	PeakMin(該 data 比之前已輸出的數值都小)

註：每次模擬只會固定選擇某一種應用功能，模擬過程中不會任意切換另一種功能。

2.3.3 IOTDF 電路的七種應用功能的運算方法與輸出方式

2.3.3.1 F1：Max(N)

本功能為比較 8 筆 IoT 資料何者最大，並將每一回合(Round)找到的最大數值作輸出，即為所求。

範例：

假設連續16筆
IoT Data輸入
如右

34	F2	77	62	32	D9	15	CF
57	D2	DC	13	68	49	F0	A5

Round_0
比較之數值

34	F2	77	62	32	D9	15	CF
----	----	----	----	----	----	----	----



輸出
(Round_0)

F2

Round_1
比較之數值

57	D2	DC	13	68	49	F0	A5
----	----	----	----	----	----	----	----



輸出
(Round_1)

F0

因此本範例之輸出順序： F2、F0、.....

註：為方便舉例，在此僅用 8bits 作範例，**實際比賽 IoT Data 為 128bits。**

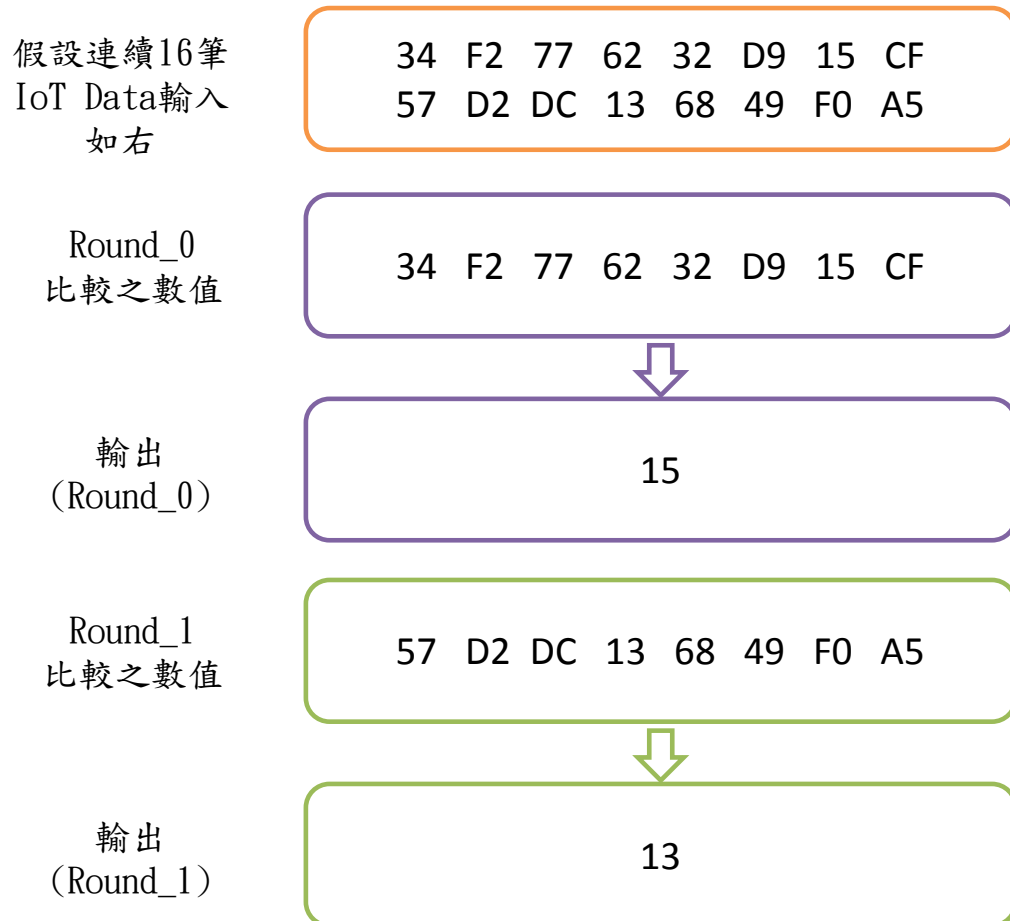
註：每一回合比完的 8 筆數值，下一回合不會再使用。

註：本題實際有 96 筆 IoT 資料，每回合必有一筆最大值輸出，因此整個模擬結果共計 12 筆輸出。

2.3.3.2 F2 : Min(N)

本功能為比較 8 筆 IoT 資料何者最小，並將每一回合找到的最小數值作輸出，即為所求。

範例：



因此本範例之輸出順序： 15、13、.....

註：為方便舉例，在此僅用 8bits 作範例，**實際比賽 IoT Data 為 128bits**。

註：每一回合比完的 8 筆數值，下一回合不會再使用。

註：本題實際有 96 筆 IoT 資料，每回合必有一筆最小值輸出，因此整個模擬結果共計 12 筆輸出。

2.3.3.3 F3 : Avg(N)

本功能係將 8 筆 IoT 資料作平均值計算，並將每一回合計算出的平均值作輸出，即為所求。

範例：

假設連續16筆
IoT Data輸入
如右

34	F2	77	62	32	D9	15	CF
57	D2	DC	13	68	49	F0	A5

Round_0
計算之數值

34	F2	77	62	32	D9	15	CF
----	----	----	----	----	----	----	----



輸出
(Round_0)

$$(34+F2+77+62+32+D9+15+CF)/8 = 7D$$

Round_1
計算之數值

57	D2	DC	13	68	49	F0	A5
----	----	----	----	----	----	----	----



輸出
(Round_1)

$$(57+D2+DC+13+68+49+F0+A5)/8 = 8B$$

因此本範例之輸出順序： 7D、8B、.....

註：為方便舉例，在此僅用 8bits 作範例，**實際比賽 IoT Data 為 128bits。**

註：每一回合計算過的 8 筆數值，下一回合不會再使用。

註：本題實際有 96 筆 IoT 資料，每回合必有一筆平均值輸出，因此整個模擬結果共計 12 筆輸出。

註：計算平均值若無法整除，僅需計算至整數位數，小數的部分採無條件捨去法。

2.3.3.4 F4：Extract(low<data<high)

本功能為比較IoT 資料有無大於已知的low 數值”且”小於已知的high 數值，若有便將其輸出，即為所求。

範例：

假設連續16筆
IoT Data輸入
如右

34 F2 77 62 32 D9 15 CF
57 D2 DC 13 68 49 F0 A5

假設
設定

low: 30
high: 70

Round_0
8筆數值

34 F2 77 62 32 D9 15 CF



輸出
(Round_0)

34、62、32

Round_1
8筆數值

57 D2 DC 13 68 49 F0 A5



輸出
(Round_1)

57、68、49

因此本範例之輸出順序： 34、62、32、57、68、49、.....

註：為方便舉例，在此僅用 8bits 作範例，**實際比賽 IoT Data 為 128bits。**

註：每一回合使用過的 8 筆數值，下一回合不會再使用。

註：有滿足這區間的 IoT 數值才會輸出，因此”未必”每個回合都有數值可輸出。

註：實際比賽，F4：Extract 功能之 low、high 數值已設為定值如下：

low: 128'h **6**FFF_FFFF_FFFF_FFFF_FFFF_FFFF_FFFF_FFFF
high: 128'h **A**FFF_FFFF_FFFF_FFFF_FFFF_FFFF_FFFF_FFFF

2.3.3.5 F5 : Exclude(data<low , high<data)

本功能為比較IoT 資料有無小於已知的low 數值”或”大於已知的high 數值，若有便將其輸出，即為所求。

範例：

假設連續16筆
IoT Data輸入
如右

34 F2 77 62 32 D9 15 CF
57 D2 DC 13 68 49 F0 A5

假設
設定

low: 30
high: 70

Round_0
8筆數值

34 F2 77 62 32 D9 15 CF



輸出
(Round_0)

F2、77、D9、15、CF

Round_1
8筆數值

57 D2 DC 13 68 49 F0 A5



輸出
(Round_1)

D2、DC、13、F0、A5

因此本範例之輸出順序： F2、77、D9、15、CF、D2、DC、13、F0、A5、.....

註：為方便舉例，在此僅用 8bits 作範例，**實際比賽 IoT Data 為 128bits。**

註：每一回合使用過的 8 筆數值，下一回合不會再使用。

註：有滿足這條條件的 IoT 數值才會輸出，因此”未必”每個回合都有數值可輸出。

註：實際比賽，F5：Exclude 功能之 low、high 數值已設為定值如下：

low: 128'h **7**FFF_FFFF_FFFF_FFFF_FFFF_FFFF_FFFF_FFFF
high: 128'h **B**FFF_FFFF_FFFF_FFFF_FFFF_FFFF_FFFF_FFFF

2.3.3.6 F6：PeakMax(該 data 比之前已輸出的數值都大)

本功能係先比較**第一回合的 8 筆 IoT 資料何者最大，先將其作輸出**，後續的回合中找到的最大值，假若有超過之前的最大值，新的最大值才需作輸出，反之，則不必輸出。

範例：

假設連續16筆
IoT Data輸入
如右

34 F2 77 62 32 D9 15 CF
57 D2 DC 13 68 49 F0 A5

Round_0
比較之數值

34 F2 77 62 32 D9 15 CF



輸出
(Round_0)

F2

Round_1
比較之數值

57 D2 DC 13 68 49 F0 A5



輸出
(Round_1)

該回合不必輸出 (因 $F0 < F2$)

因此本範例之輸出順序： F2、.....

本範例中，後續各回合找到的最大值若有超過 F2，才會再輸出。

註：為方便舉例，在此僅用 8bits 作範例，**實際比賽 IoT Data 為 128bits**。

註：每一回合使用過的 8 筆數值，下一回合不會再使用。

註：**各回合中，有超過之前已輸出的最大值才會輸出，因此”未必”每個回合都有數值可輸出。**

2.3.3.7 F7：PeakMin(該 data 比之前已輸出的數值都小)

本功能係先比較**第一回合的 8 筆 IoT 資料何者最小**，**先將其作輸出**，後續的回合中找到的最小值，假若有低於之前的最小值，新的最小值才需作輸出，反之，則不必輸出。

範例：

假設連續16筆
IoT Data輸入
如右

34	F2	77	62	32	D9	15	CF
57	D2	DC	13	68	49	F0	A5

Round_0
比較之數值

34	F2	77	62	32	D9	15	CF
----	----	----	----	----	----	----	----



輸出
(Round_0)

15

Round_1
比較之數值

57	D2	DC	13	68	49	F0	A5
----	----	----	----	----	----	----	----



輸出
(Round_1)

13 (因 13 < 之前輸出的15)

因此本範例之輸出順序： 15、13、.....

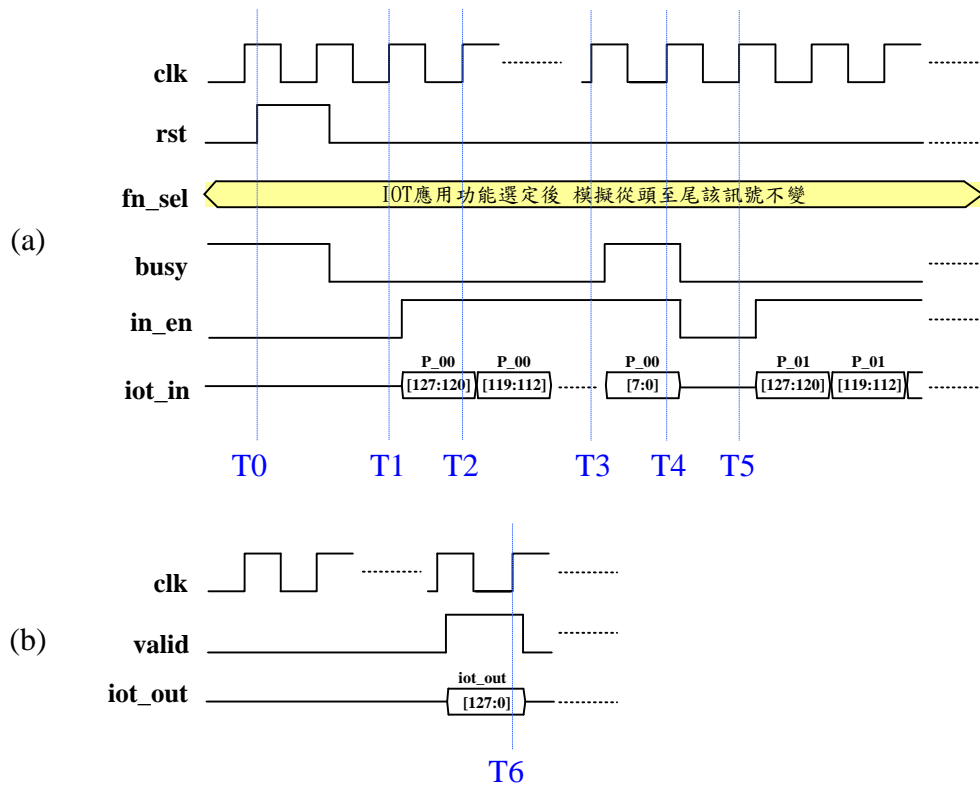
本範例中，後續各回合找到的最小值若有低於 13，才會再輸出。

註：為方便舉例，在此僅用 8bits 作範例，**實際比賽 IoT Data 為 128bits**。

註：每一回合使用過的 8 筆數值，下一回合不會再使用。

註：各回合中，有低於之前已輸出的最小值才會輸出，因此”未必”每個回合都有數值可輸出。

2.4 IOTDF 電路時序規格



圖四、(a) IOTDF 電路輸入之時序圖、(b) IOTDF 電路輸出之時序圖

1. T0 ~ T1 時間點，IOTDF 電路初始化。
2. T1 時間點，Host 端判斷 busy 訊號為 low，隨即 in_en 拉為 high，同時輸入 IoT Data P_00[127:120] 這區間的資料。
3. T2 時間點，Host 端判斷 busy 訊號為 low，in_en 繼續維持 high，繼續輸入 IoT Data 下一個區間 P_00[119:112] 的資料，T1~T2 這區間即為連續資料輸入的範例。
4. T3 時間點，Host 端判斷 busy 訊號依舊為 low，in_en 繼續維持 high，繼續輸入 IoT Data 最後一個區間 P_00[7:0] 的資料。
5. T4 時間點，Host 端判斷 busy 訊號為 high，隨即 in_en 為 low，IoT Data 此時輸入為 0，表示暫停資料輸入。
6. T1~T4 這區間為輸入完整 1 筆 IoT Data 的範例，共計花費 16 個 Cycles，這過程中若有需要暫停輸入，可以自行將 busy 訊號拉為 high。
7. T5 時間點以後，請自行重覆 T1 ~ T4 動作之行為，即可讓 Host 端繼續輸入下一筆 IoT Data P_01 的資料。在 IoT 資料輸入期間，參賽者可以一邊輸入一邊根據指定應用功能作運算或是一個回合的 8 筆資料都齊全再行運算也可以，請自行決定。
8. T6 時間點，當 IOTDF 電路運算過程需要作輸出時，請自行將 valid 訊號拉為 high，同時將欲輸出值透過 iot_out[127:0] 直接輸出即可。

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(Time)**、**平均功率消耗(Power)**，主辦單位會依此兩項目做為同等級之評分。由於**本題目標”Power 越低者”離”特優”腳步越近，但 PrimeTimePX 的 Power Report 是平均功率消耗，為防止有同學故意拉長模擬時間製造低的平均功率消耗，因此 Score = Power x Time，即可得知各 Functions 實際消耗的焦耳數(J)是多少。**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圖五範例，紀錄成 **Time =16590 ns** 做評分。

```
ncsim> run
FSDB Dumper for IUS, Release Verdi3_L-2016.06-SP1-1, Linux, 09/27/2016
(C) 1996 - 2016 by Synopsys, Inc.
*Verdi3* : Create FSDB file 'IOTDF_F3.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.
-----

Start to Send IOT Data & Compare ...

P00:  ** Correct!!  **
P01:  ** Correct!!  **
P02:  ** Correct!!  **
P03:  ** Correct!!  **
P04:  ** Correct!!  **
P05:  ** Correct!!  **
P06:  ** Correct!!  **
P07:  ** Correct!!  **
P08:  ** Correct!!  **
P09:  ** Correct!!  **
P10:  ** Correct!!  **
P11:  ** Correct!!  **

-----

Congratulations! All data have been generated successfully!
-----PASS-----

Simulation complete via $finish(1) at time 16590 NS + 0
./testfixture.v:227      #(`CYCLE/2); $finish;
```

圖五 Gate-level Simulation Example

註：本題因有 7 種應用功能，因此會有 7 種 Simulation Time。

✧ 評分項目二：依”功率消耗”(Power)大小評分

各參賽隊伍將 APR 完成後，請用 PrimeTime-PX (後文以 PT-PX 表示)作 Power 分析，本題主辦單位已提供 PT-PX 的自動化執行程式：`pt_script.tcl`，參賽者只要將 `pt_script.tcl` 檔案會使用到的相關檔案(即 `IOTDF_pr.v`、`IOTDF_APR.sdc`、`IOTDF.fsdb`、`.synopsys_pt.setup`)準備好，執行下述指令即可作 Power 分析，執行後之 log 訊息如下所示，功率消耗請自行轉換單位為毫瓦(mW)，以此例讀作 0.8994 mW，紀錄成 `Power=0.8994` 作評分。

```
unix% pt_shell -f ./pt_script.tcl
```

Power Group	Internal Power	Switching Power	Leakage Power	Total Power	(%)	Attrs
clock_network	5.462e-04	1.347e-04	1.455e-06	6.824e-04	(75.88%)	i
register	9.190e-05	4.246e-05	2.589e-05	1.603e-04	(17.82%)	
combinational	1.589e-05	3.080e-05	1.001e-05	5.670e-05	(6.30%)	
sequential	0.0000	0.0000	0.0000	0.0000	(0.00%)	
memory	0.0000	0.0000	0.0000	0.0000	(0.00%)	
io_pad	0.0000	0.0000	0.0000	0.0000	(0.00%)	
black_box	0.0000	0.0000	0.0000	0.0000	(0.00%)	

Net Switching Power	= 2.080e-04	(23.13%)				
Cell Internal Power	= 6.540e-04	(72.72%)				
Cell Leakage Power	= 3.736e-05	(4.15%)				

Total Power	= 8.994e-04	(100.00%)				

X Transition Power	= 1.331e-05					
Glitching Power	= 1.149e-06					

Peak Power	= 0.0857					
Peak Time	= 15665.999					

註：本題因有 7 種應用功能，因此會有 7 種 Power 數值。

註：未完成 APR 者，請以合成後結果作為等級 C 之評分依據。

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

- 功能正確，RTL 模擬與正確解答比對完全正確。
- 完成 Synthesis，且 Gate-Level Pre-layout Simulation 結果正確。
- 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

- 只需做 Marco layout (即不用包含 IO Pad、Bonding Pad)。
- VDD 與 VSS Power Ring 寬度請各設定為 **2um**，只須做一組。
- 不需加 Dummy Metal。
- 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：

$$\text{Score} = \text{Power1} \times \text{Time1} + \text{Power2} \times \text{Time2} + \dots + \text{Power7} \times \text{Time7}$$

註：請自行將 7 種 Function 的各 Score 值作相加，當作最後分數。

註：Score 越小者，同級名次越好！

註：達到等級 A，卻 Score 與同級相比不夠低者，未必可得獎，請注意功率消耗！

- ◇ 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：

$$\text{Score} = (\text{Power1} \times \text{Time1} + \text{Power2} \times \text{Time2} + \dots + \text{Power7} \times \text{Time7}) \times (\text{DRC} + \text{LVS 的總錯誤量})$$

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：

$$\text{Score} = \text{Power1} \times \text{Time1} + \text{Power2} \times \text{Time2} + \dots + \text{Power7} \times \text{Time7}$$

註：

1. Score 越小者，同級名次越好！
2. 等級 C，視 APR 為 Fail，Time 以 Gate-level Pre-layout Simulation 為主。
3. 等級 C，視 APR 為 Fail，Power 以合成後的相關檔案使用 PT-PX 作 Power 分析。

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

$$\text{Score} = \text{Function1 errors} + \text{Function2 errors} + \dots + \text{Function7 errors}$$

註：

1. 等級 D，Score 評分方式為 7 種 Functions 各個模擬的 error 總數作相加。
2. 等級 D，視合成與 APR 皆為 Fail，Power、Time 將不予考慮。
3. 等級 D，只以 RTL Simulation 正確率為主，Score 越小者(即 error 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

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
IOTDF.v	本題之設計檔，已包含系統 Input/Output Port 之宣告，請以此檔案作為 IOTDF 電路之設計。
testfixture.v	本題僅有一個 TestBench，卻有 7 種 Functions 需模擬，因此請在模擬期間，自行使用 +define+F1、+define+F2、...、+define+F7 參數，作 7 種 Functions 模擬時的切換。
pattern1.dat	IOTDF 電路模擬時輸入的 96 筆 IoT Data，7 種 Functions 皆使用同一組測試樣本。 註：這些檔案已加入至 TestBench，無需額外設定。
f1.dat ~ f7.dat	作為 IOTDF 電路模擬時，7 種 Functions 的輸出值與標準解答作比對。 註：這些檔案已加入至 TestBench，無需額外設定。
IOTDF_DC.sdc	Design Compiler 作合成之 Constraint 檔案，請自行設定 period 的期望值，但環境相關參數請勿更改。
IOTDF_APR.sdc	Innovus、IC Compiler 作 APR 之 Constraint 檔案，請自行設定 period 的期望值，但環境相關參數請勿更改。
.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。 註：無論合成或 APR，只需使用 worst case library。

.synopsys_pt.setup	使用 PT-PX 作 Power 量測之初始化設定檔。參賽者請依 Library 實際擺放位置，自行修改 Search Path 的設定。 註：作 Power 量測，只需使用 worst case library。
pt_script.tcl	使用 PT-PX 作 Power 量測之自動化執行 Script，參賽者使用前，請自行將會用到的相關檔案擺在目前目錄下，方可正常執行。
runall_rtl	本題提供 7 種應用功能作 RTL-Simulation 批次檔。以方便參賽者快速模擬驗證。
runall_syn	本題提供 7 種應用功能作 Pre-layout Gate-level Simulation 批次檔。以方便參賽者快速模擬驗證。
runall_pr	本題提供 7 種應用功能作 Post-layout Gate-level Simulation 批次檔。以方便參賽者快速模擬驗證。

2. 本次比賽中，會有隱藏的 Pattern 作測試，其 IoT Data 僅是比賽用的數值作順序調換。

3. 使用 Innovus 作 APR 請注意，模擬時請務必自行加上**+ncmaxdelays** 參數。

例如：

```
> ncverilog +ncmaxdelays testfixture.v IOTDF_pr.v tsmc13_neg.v +define+F1+SDF
+access+r
```

註：本次比賽中有 7 種 Functions，請自行使用+define+F1、+define+F2、...、 +define+F7 參數作切換。

附錄 B 測試樣本

本題的 7 種應用功能皆使用同一組測試樣本，共計 96 筆 IoT Data，如下：

```
1A_A9_92_A8_74_2B_E4_86_B7_89_00_B6_8F_C1_38_1D //Round_00 P_00: 3.544040e+37
D1_99_D7_A0_25_9F_50_AA_B8_CA_1A_89_78_5D_99_5B //Round_00 P_01: 2.786074e+38
FA_0F_13_92_4B_6C_48_75_BD_5F_5E_96_5C_8C_6F_C9 //Round_00 P_02: 3.323853e+38
C0_E4_42_33_61_AD_10_37_B0_C0_EA_36_5F_38_2C_4E //Round_00 P_03: 2.563970e+38
E8_F4_B0_68_EC_2D_0F_38_42_A7_FF_69_62_41_31_BB //Round_00 P_04: 3.096514e+38
9B_2E_6E_48_31_E3_94_E3_02_8D_2B_84_41_BC_30_A5 //Round_00 P_05: 2.062714e+38
75_12_52_D5_A7_28_D6_4D_59_D9_2C_D6_F8_50_EE_D6 //Round_00 P_06: 1.556148e+38
0B_FC_0B_C1_64_19_58_40_E5_6C_1A_A8_A9_6C_7D_85 //Round_00 P_07: 1.593021e+37
99_6D_4F_DD_31_C2_59_E5_BE_93_9B_7B_81_BF_CB_61 //Round_01 P_00: 2.039395e+38
C8_C2_6D_6A_08_E4_0A_1E_4B_C4_69_DA_32_5A_2F_9E //Round_01 P_01: 2.668551e+38
D1_EC_9D_BE_CD_3F_64_AA_2F_8A_41_7B_1A_72_3A_9A //Round_01 P_02: 2.790372e+38
39_DF_3E_11_50_98_DD_53_29_FB_33_45_E3_08_D9_89 //Round_01 P_03: 7.692514e+37
BB_F6_D8_F5_BF_13_06_53_13_1B_30_BB_5C_BA_7E_BA //Round_01 P_04: 2.498473e+38
AD_4B_E1_AF_97_44_2B_D3_6F_4D_E2_F8_74_9F_FC_60 //Round_01 P_05: 2.303504e+38
F8_B5_ED_42_4B_D8_D4_8C_45_61_A1_60_10_20_2C_1F //Round_01 P_06: 3.305932e+38
69_E6_90_BC_D0_AF_03_A5_3C_79_9F_3D_D9_37_F3_5D //Round_01 P_07: 1.407661e+38
0D_DF_4D_7A_E4_7C_77_D8_25_38_A8_3E_47_4E_61_CF //Round_02 P_00: 1.843942e+37
50_D4_C5_DF_96_5B_D4_98_78_52_2D_04_E9_07_72_E5 //Round_02 P_01: 1.074430e+38
50_09_81_52_D0_36_E6_54_BA_3A_A0_FB_0F_33_5C_58 //Round_02 P_02: 1.063876e+38
```

pattern1.dat

註：左側為 128bits IoT Data，以十六進制表示，資料右側以註解方式分別說明 IoT Data 是：哪一回合(Round)、第幾個 Pattern、及等值於 10 進制的大概值。

註：每一回合的 IoT Data 筆數皆為 8 個。

本電路的 7 種應用功能，計算出之標準答案，分別存放於 f1.dat ~ f7.dat 檔案之中，例如：下圖為第 1 種應用功能的標準解答，格式如下：

```
FA_0F_13_92_4B_6C_48_75_BD_5F_5E_96_5C_8C_6F_C9 //Round_00
F8_B5_ED_42_4B_D8_D4_8C_45_61_A1_60_10_20_2C_1F //Round_01
C8_82_4C_5D_1D_86_50_29_5A_E7_40_FE_09_64_68_C3 //Round_02
FC_EA_3F_27_9D_92_5D_C5_2E_C4_C7_EE_6C_90_5B_93 //Round_03
EA_11_44_1E_A8_23_A0_AD_E3_85_2D_25_C7_1F_75_0A //Round_04
EE_CC_FE_0E_A6_2B_02_AD_92_C0_25_E8_6B_D3_03_DB //Round_05
D4_95_A1_68_F2_98_7F_A5_E0_66_73_E0_F6_7F_60_28 //Round_06
FD_79_95_25_A5_79_3D_4C_74_D1_B8_1B_5D_F2_8D_34 //Round_07
DF_5E_9B_3E_AA_F1_DE_60_EF_68_67_2F_E6_D0_1D_CC //Round_08
E8_31_4A_E6_0F_F5_85_0B_B1_FE_CC_DF_54_9D_C7_80 //Round_09
F6_CA_50_3E_3E_CA_BF_18_8B_00_99_2F_75_A2_9E_03 //Round_10
CD_79_44_C2_00_CE_A1_8C_4E_AB_03_28_C5_44_FC_0F //Round_11
```

f1.dat

註：資料左側為十六進制，資料右側以註解方式說明該筆資料是哪一個回合的標準答案。

註：其他 6 種應用功能，請自行參閱 f2.dat ~ f7.dat 檔案。

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，模擬結果必須滿足本題指定之 Period 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、SDF 檔、SPEF 檔** 完成 post-layout gate-level simulation 及 Power Analysis，以下分為 IC Compiler、Innovus 兩種軟體說明 netlist、sdf、SPEF 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	IOTDF_pr.sdf
Significant digits	2

按 **OK**。

對應指令： write_sdf -version 2.1 IOTDF_pr.sdf

“File > Export > Write Verilog...”

先按 **Default**

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

按 **OK**。

“File > Export > Write Parasitics...”

先按 **Default**

Net Format in file	SPEF
Output file name prefix	IOTDF_pr.spef
All other options	Default value

按 **OK**。

ii. 使用 Cadence Innovus 者，執行步驟如下：

在 Innovus 視窗下點選：

“File → Save → Netlist...”

Netlist File	IOTDF_pr.v
All other options	Default value

按 。

“Timing → Extract RC...”

Save SPEF to	IOTDF_pr.spef
All other options	Default value

按 。

“Timing → Write SDF...”

Ideal Clock	Disable
SDF Output File:	IOTDF_pr.sdf

按 。

3. 完成 DRC 與 LVS 驗證：(今年起驗證方法不同，請注意)

DRC 驗證： 統一使用 Mentor Calibre DRC 作驗證

LVS 驗證： 統一使用 P&R 軟體內建之 LVS 作驗證

A. 使用 Metor Calibre DRC 作驗證

- 準備 P&R 後的 GDSII 檔案，檔名請命名為 IOTDF_pr.gds
- 準備 DRC 驗證檔案， 檔名為 Calibre-drc-cur
- 使用 Metor Calibre DRC 指令作驗證：

calibre -drc Calibre-drc-cur

d. DRC 驗證結果，目標作到如下圖，TOTAL RESULT GENERATED = 0

```
--- CALIBRE::DRC-F EXECUTIVE MODULE COMPLETED. CPU TIME = 3 REAL TIME = 3
--- TOTAL RULECHECKS EXECUTED = 701
--- TOTAL RESULTS GENERATED = 0
--- DRC RESULTS DATABASE FILE = DRC_RES.db (ASCII)

--- CALIBRE::DRC-F COMPLETED - Mon Apr 1 14:21:15 2019
--- TOTAL CPU TIME = 3 REAL TIME = 5
--- SUMMARY REPORT FILE = DRC.rep
```

B. 使用 P&R 軟體內建之 LVS 作驗證

以下分為 IC Compiler、Innovus 兩種軟體說明執行步驟。

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

在 IC Compiler Layout 視窗底下點選

“Verification > LVS ...”

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

按 。

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 Innovus 者，驗證 LVS 步驟如下：

在 Innovus 視窗下點選

請選“Verify → Verify Connectivity...” Default 值，按 。

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

4. Macro Layout 產生 IO Pins 的作法 (限 Innovus 參賽者)

在 Innovus 視窗下點選

請選“File → Save → I/O File...”

Save IO	sequence
To File	IOTDF.io
Generate template IO File	enable

按 。

請選“File → Load → I/O File...”

請點選 IOTDF.io，按 。

即可！

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 Innovus 者，請將 Innovus 相關的 design database (包含 .enc 檔案與 .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 3

RTL category		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
N/A	N/A	Design Report Form
RTL Simulation	*.v or *.sv	Verilog 或 System Verilog ...
Gate-Level category		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
Pre-layout Gate-level Simulation	IOTDF_syn.v	Verilog gate-level netlist generated by Synopsys Design Compiler
	IOTDF_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	IOTDF_pr.v	Verilog gate-level netlist generated by Cadence Innovus or Synopsys IC Compiler
	IOTDF_pr.sdf	Post-layout gate-level sdf
	IOTDF_pr.spef	Post-layout Extract RC - SPEF

附錄 E 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 IOTDF.v ~/result/**
 - > **cp IOTDF_pr.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. 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
	PrimeTime-PX	pt_shell
	Verdi *1	verdi, nWave, nLint
Mentor	Calibre *3	calibre
	QuestaSim	vsim
Utility	vi	vi, vim
	gedit	gedit
	nedit	nedit
	pdf reader	acroread
	calculate	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

附錄 G 設計資料庫

設計資料庫位置： /usr/cad/icc2019/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
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
--------------	--------------------------

Calibre /

Calibre-drc-cur

Verify DRC file

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 (例如：Verilog、System Verilog)	
RTL Simulation	RTL 檔案名稱 (RTL 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 Innovus)	
	設計資料庫檔案名稱 (Library name) (ICC: Milkyway Library Name, Innovus: xxx.enc.dat)	
	Calibre DRC 錯誤總數量 (ex: 0 個)	
	APR Tool LVS 錯誤總數量 (ex: 0 個)	
	Post-layout Simulation 所使用的 CYCLE Time (ex: 10ns) (請確定模擬功能正確)	
Score (Pre-layout or Post-layout Result) Over All	Simulation Time (ex: time = 16590 ns) time = ? (7 種應用功能都寫)	
	Power Measure using PrimeTime-PX (ex: power = 0.8994 mW) power = ? (7 種應用功能都寫)	
	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		