2018 IC Design Contest Final

大學類標準元件數位電路設計

Two Protocol Arbiter for Register Spaces Access

1. 問題描述

本題請完成一雙協定暫存器存取仲裁電路 (Two Protocol Arbiter for Register Spaces Access, 後文以 TPA 表示)。其電路架構如圖 1.所示，TPA 電路含有一暫存器空間陣列(256x16, 大小為 256 個 word address，每個 word 由 16bits 組成)，Testbench 電路將發送兩種不同協定的控制訊號到 DUT 電路內，這兩種協定分別是 Two-Wire Protocol(後文以 TWP 表示)和 Register Interface Master(後文以 RIM 表示)，參賽隊伍必須設計一仲裁器來操作這兩個協定對暫存器的讀取與寫入；而 TWP 又可分為位於 Testbench 內的 Two-Wire Protocol Master (TWM)電路及位於 TPA 內 Two-Wire Protocol Slaver (TWS)電路，由於 TWP 為一雙線串列資料傳輸協定，因此參賽者必須在 TPA 內設計一 TWS 資料轉換電路以將串列傳輸資料轉成可用於暫存器陣列存取的格式。

本次 IC 設計競賽比賽時間為上午 08:30 到下午 08:30。當 IC 設計競賽結束後，CIC 會根據第 3 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 中所列的要求，附上評分所需要的檔案。

軟體環境及設計資料庫說明請參考附錄 A 與附錄 E。

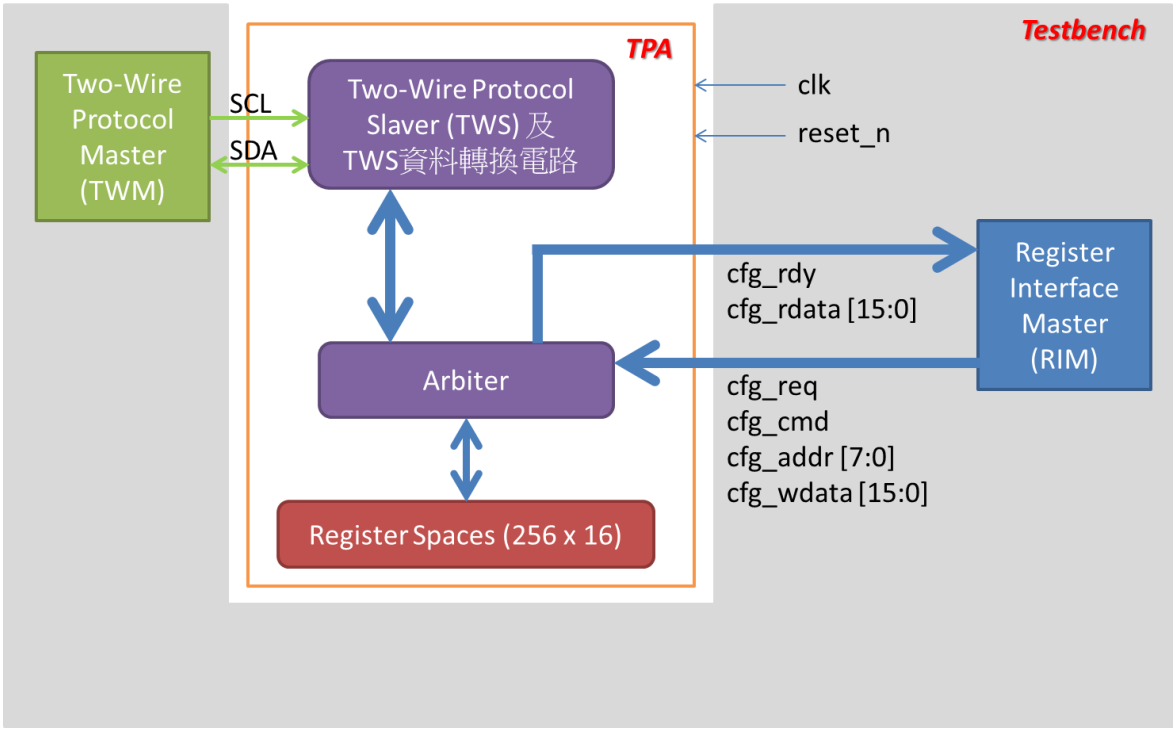


圖 1. 雙協定暫存器存取系統架構

本題目之測試樣本置於 */usr/cad/icc2018/euc/icc2018cb.tar*，請執行以下指令取得測試樣本：

tar xvf /usr/cad/icc2018/euc/icc2018cb.tar

2. 設計規格

2.1 系統方塊圖

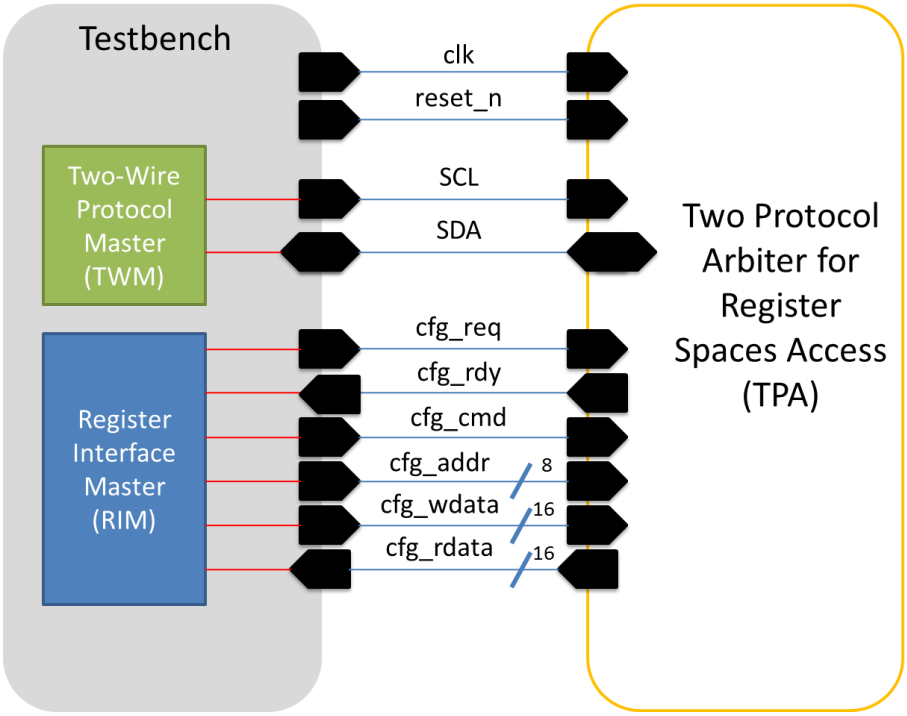


圖 2. 系統方塊圖

2.2 輸出入訊號和記憶體描述

表一、輸入/輸出信號

Signal Name	I/O	Width	Simple Description
clk	I	1	系統時脈訊號。本系統為同步於時脈**正緣**之同步設計。
reset_n	I	1	**低**位準”非”同步(active low asynchronous)之系統重置信號。
SCL	I	1	TWS 電路時脈訊號。本電路為同步於時脈**正緣**之同步設計。
SDA	IO	1	TWS 電路資料傳輸及接收訊號。 此訊號為雙向(bidirectional)設計，Testbench 內連接一只 pull-up resistor 在此訊號上，故當此訊號兩端沒有任何訊號驅動時，將呈現 High 狀態。 **使用上特別注意，不可讓此訊號兩端同時驅動，否則會出現非預期的結果(unknown)。**

cfg_req	I	1	RIM 動作要求訊號。 當此訊號為 High 時，表示 RIM 要求進行寫入或讀取暫存器空間動作。
cfg_rdy	O	1	RIM 動作完成訊號。 當 cfg_req 及 cfg_rdy 同時為 High 時，表示 TPA 已經接收到 cfg_req 所要求的指令，當 cfg_rdy 再次變成 Low 時表示指令已經完成；故當 Testbench 偵測到 cfg_rdy 再變成 Low，表示 Testbench 可以進行下一筆資料的存取。
cfg_cmd	I	1	RIM 指令訊號。 當 cfg_cmd 為 1，表示要進行寫入。 當 cfg_cmd 為 0，表示要進行讀取。
cfg_addr	I	8	RIM 位址訊號。 指示此一動作將對哪個暫存器空間位址進行寫入或讀取資料。
cfg_wdata	I	16	RIM 寫入資料訊號。 指示此一動作將寫入哪些資料到暫存器空間位址。
cfg_rdata	O	16	RIM 讀取資料訊號。 指示此一動作將由暫存器空間位址讀取哪些資料。

2.3 系統功能描述

當 reset_n 啟動結束後，Testbench 會開始利用 TWP 協定規格及 RIM 協定規格**分別或同時**發送**讀取或寫入**資料到 TPA 電路，TPA 電路功能符合以下規格:

1. TWP 及 RIM 協定都可對 Register spaces 進行讀取及寫入動作。
2. TWP 及 RIM 協定可**同時**對 Register spaces 進行讀取動作。
3. TWP 及 RIM 協定在**同一時間**對**同一暫存器空間(位址)**進行寫入時，以 RIM 的動作為主；TWP 的動作將被忽略。
4. TWP 及 RIM 協定在**同一時間**對**不同暫存器空間(位址)**進行寫入時，則這兩個協定的動作都會被進行。
5. TWP 及 RIM 協定在**不同時間**對**同一暫存器空間(位址)**進行寫入時，則這兩個協定的動作都會被進行。本題需考慮當 RIM 指令觸發後，下一個時脈週期就發生 TWP 指令觸發的情況；及須考慮 TWP 指令觸發後的下一時脈週期就發生 RIM 指令觸發的情況。
6. RIM 使用 clk 做為時脈訊號，TWP 使用 SCL 作為時脈訊號；clk 與 SCL 為完全相同的時脈訊號。

當 TPA 收到 RIM 或 TWP 的指令後，需依照指令指示將資料讀出或寫入 TPA 內建的電路暫存器空間陣列(已宣告於 TPA.v 電路內，Verilog code 如下，參賽者不可更改此宣告，否則將造成電路誤動作)。

```
reg       [15 : 0]       Register_Spaces       [0 : 255];
```

Testbench 將會依照以下幾個 Stage 進行測試。Testbench 會在每個 Stage 測試結束後，進行資料比對。

Stage1.：利用 RIM 依序將待測向量的第 0 ~ 31 筆資料**寫入**暫存器空間陣列位址 0 ~ 位址 31。(動作時序圖請參考 2.4.1)

Stgae2.：利用 RIM 依序將暫存器空間陣列的第 0 ~ 31 筆資料**讀出**，再依序**寫入**暫存器空間陣列位址 63 ~ 位址 32。(動作時序圖請參考 2.4.1)

Stage3.：利用 TWP 依序將待測向量的第 64 ~ 95 筆資料**寫入**暫存器空間陣列位址 64 ~ 位址 95。(動作時序圖請參考 2.4.2)

Stage4.：利用 TWP 依序將暫存器空間陣列的第 64 ~ 95 筆資料**讀出**，再依序**寫入**暫存器空間陣列位址 127 ~ 位址 96。(動作時序圖請參考 2.4.2)

Stage5.：將 RIM 及 TWP 同時執行動作測試，共有兩個測試項。第一個測試項，利用 RIM 將待測向量的第 *i* 筆資料寫入暫存器空間陣列位址 *i* ；同時利用 TWP 將待測向量的第 *255-i* 筆資料寫入暫存器空間陣列位址 *255-i* 。接著換第二個測試項，利用 RIM 將暫存器空間陣列位址 *i* 的資料讀出，再寫入暫存器空間陣列的 *i+1* ；同時利用 TWP 將暫存器空間陣列位址 *255-i* 的資料讀出，再寫入暫存器空間陣列的 *255-(i+1)*，接著將 *i* 累加 1 後回到第一個測試項再次進行測試。

Stage6.：將針對同一位址進行寫入動作，共分三個測試項。6-1 測試項將會讓 RIM 及 TWP 在同一時脈週期指令觸發，由於 RIM 為 master 地位，故此情況只需進行 RIM 指令所要求的動作即可，TWP 指令將被忽略。6-2 測試項將會讓 RIM 先行指令觸發，而 TWP 指令觸發將會出現在下一個時脈週期，因此暫存器空間陣列最終將顯示 TWP 所要求的指令結果(RIM 的指令動作也可能暫時出現在暫存器空間陣列中，但此不在評分範圍內)。6-3 測試項會讓 TWP 先指令觸發後，下一時脈週期出現 RIM 指令觸發的情況，因此暫存器空間陣列最終將顯示 RIM 所要求的指令結果(TWP 的指令動作也可能暫時出現在暫存器空間陣列中，但此不在評分範圍內)。

關於上述 RIM 及 TWP 的指令觸發請參考動作時序圖 2.4.1 及 2.4.2 說明。

2.4 系統時序規格圖

系統控制訊號時序規格圖，如下圖 3.所示。

圖 3. 系統控制訊號時序規格

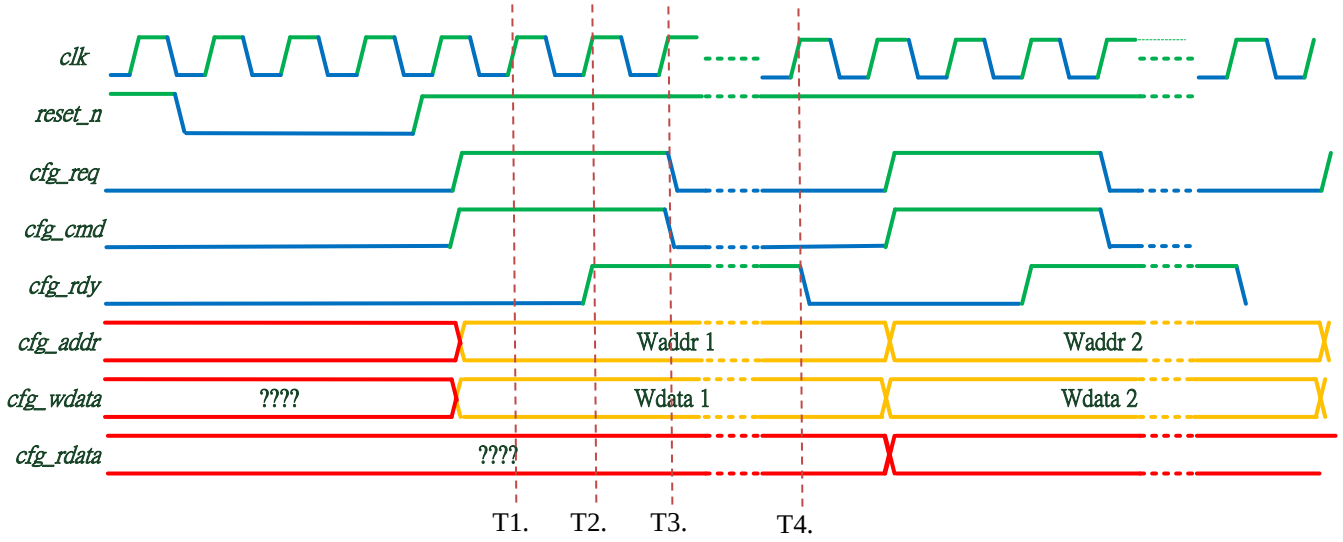
a、 本電路的 reset_n 訊號為低位準啟動，*reset_n* 訊號持續 3 個時脈週期後結束。

b、 reset_n 訊號動作結束後，Testbench 電路就會開始發出 RIM 或 TWP 的資料存取指令。關於 RIM 電路的資料存取說明及時序請參考 2.4.1；關於 TWP 電路的資料存取說明及時序請參考 2.4.2。

c、 本題會進行 6 個 Stage 的測試，當每一個 Stage 的測試項目完成後，Testbench 會立即比對 TPA 內的暫存器空間陣列 Register_Spaces 的內容來確認此 Stage 的模擬結果。當 6 個 Stage 的測試項都完成後，模擬會立即結束。

2.4.1 RIM 電路訊號說明及時序規格

✓ 利用 RIM 協定進行資料寫入暫存器空間陣列



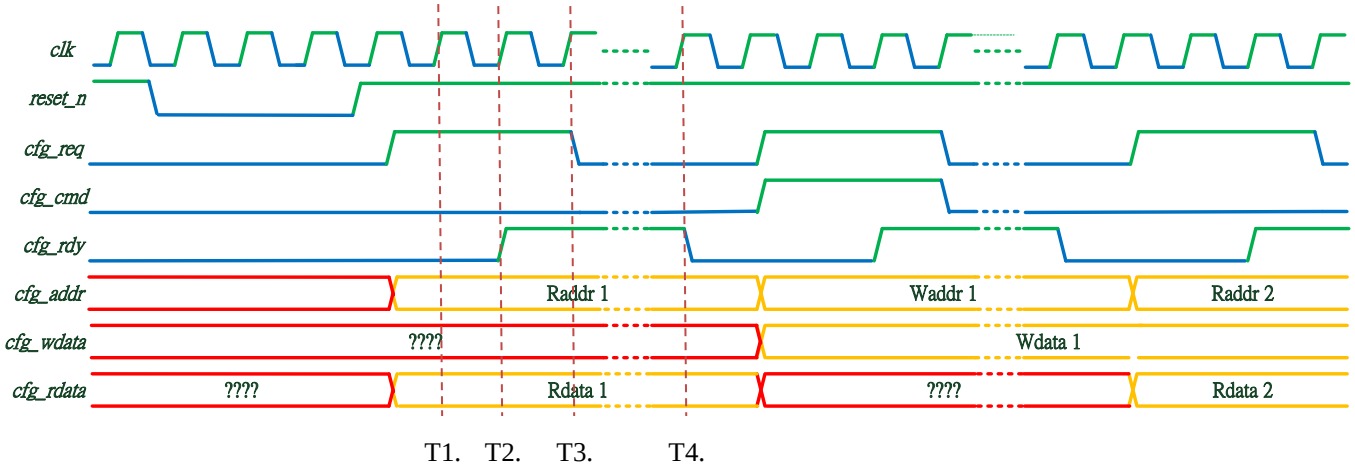
T1： 當時脈週期正緣觸發時，若 cfg_req 為 1 表示為指令觸發；此時 Testbench 會將 cfg_cmd 設為 1 表示進行資料寫入，並將 cfg_addr 位址及 cfg_wdata 資料送出。

T2： 當 TPA 收到指令觸發後，必須在下個時脈週期之正緣將 cfg_rdy 設為 1 表示收到指令。同時開始進行資料寫入動作。

T3： 當 Testbench 收到 cfg_rdy 輸出為 1 後，就會在下個時脈週期之正緣將 cfg_req 設為 0，表示指令已經送達 TPA。

T4： 當寫入動作尚未完成時，將 cfg_rdy 維持在 1，cfg_rdy 為 1 可以維持 1 個或數個時脈週期；當 TPA 完成資料寫入動作後，將 cfg_rdy 設為 0，表示完成這次的指令；。而 Testbench 收到 cfg_rdy 輸出為 0 時，表示可進行下一筆資料的存取指令。

✓ 利用 RIM 協定進行資料讀取及寫入暫存器空間陣列



T1：當時脈週期正緣觸發時，若 cfg_req 為 1 表示為指令觸發；此時 Testbench 會將 cfg_cmd 設為 0 表示進行資料讀取，並同時將 cfg_addr 位址送出。

T2：當 TPA 收到指令觸發後，必須在下個時脈週期之正緣將 cfg_rdy 設為 1 表示收到指令。同時開始進行資料讀取動作。

T3：當 Testbench 收到 cfg_rdy 輸出為 1 後，就會在下個時脈週期之正緣將 cfg_req 設為 0，表示指令已經送達 TPA。

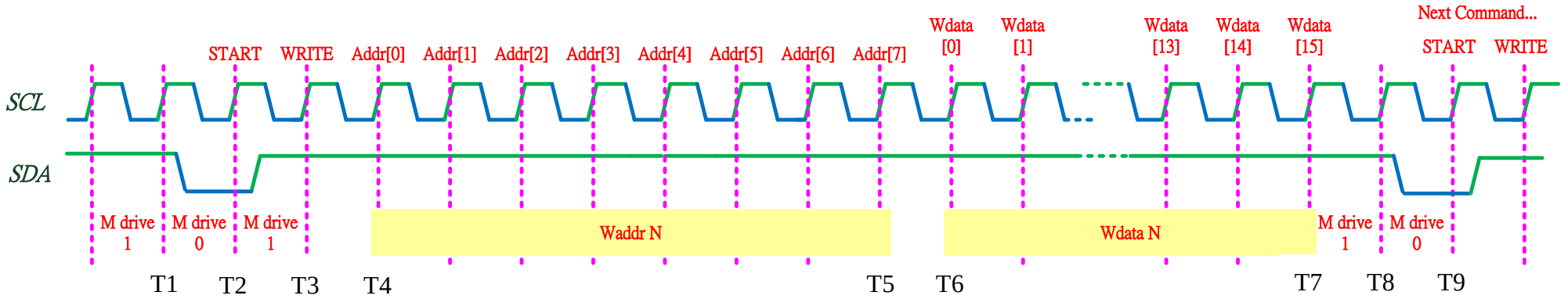
T3~T4：在這區間，testbench 會在第一個時脈週期**負緣**且 cfg_rdy 為 1 的瞬間將 TPA 讀出的資料進行取樣。

T4：當資料讀取尚未完成時，將 cfg_rdy 維持在 1，cfg_rdy 為 1 可以持續 1 個或數個時脈週期；當 TPA 完成資料讀取動作後，將 cfg_rdy 設為 0，表示完成這次的指令。而 testbench 收到 cfg_rdy 輸出為 0 時，表示可進行下一筆資料的存取指令。

2.4.2 TWP 電路訊號說明及時序規格

以下圖中的”M drive 1”表示由 TWM 將 SDA 設定為 1，”M drive 0”表示由 TWM 將 SDA 設定為 0。”S drive 1”表示由 TWS 將 SDA 設定為 1；”S drive 0”表示由 TWS 將 SDA 設定為 0。

✓ 利用 TWP 協定進行資料寫入暫存器空間陣列；所有動作將由 TWM 發出，TWS 負責接收及執行。



T1：一開始 TWM 將 SDA 設為 1，此時 TWP 系統為閒置的狀態。

T2：當時脈正緣時，TWM 將 SDA 設為 0 表示指令觸發，TWS 開始執行指令動作。

T3：由 SDA 位準為 1 或 0 來指示要進行何種動作。1 表示進行寫入資料；0 表示進行讀取資料。

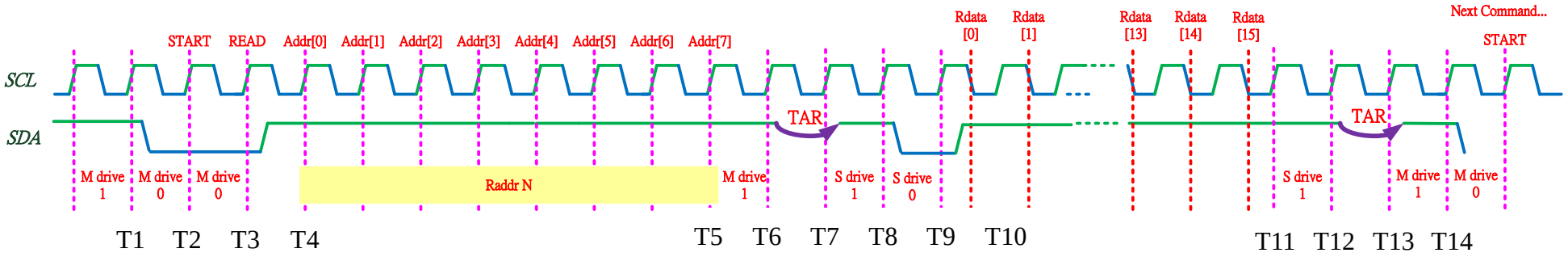
T4~T5：這段時間共有 8 個時脈週期，TWM 依序將寫入位址利用 SDA 由 LSB 開始送出。

T6~T7：這段時間共有 16 個時脈週期，TWM 依序將寫入資料利用 SDA 由 LSB 開始送出。

T8：接下來，TWM 再次將 SDA 設為 1，表示 TWP 系統再次回到閒置的狀態。等同 T1 時間。

T9：當時脈正緣時，TWM 將 SDA 設為 0 表示指令觸發，TWS 開始執行下一筆指令動作。等同 T2 時間。

✓ 利用 TWP 協定進行資料讀出暫存器空間陣列；由 TWM 發出資料讀取指令，TWS 收到指令後將讀出的資料經 SDA 送給 TWM。



T1：一開始 TWM 擁有 SDA 訊號的控制權，並將 SDA 設為 1，此時 TWP 系統為閒置的狀態。

T2：當時脈正緣時，TWM 將 SDA 設為 0 表示指令觸發，TWS 開始執行指令動作。

T3：由 SDA 位準為 1 或 0 來指示要進行何種動作。1 表示進行寫入資料；0 表示進行讀取資料。

T4~T5：這段時間共有 8 個時脈週期，TWM 依序將讀取位址利用 SDA 由 LSB 開始送出。

T6：接下來，TWM 將 SDA 設為 1 並準備將 SDA 變成 TAR 狀態。

T7：接下來 SDA 成為 TAR 狀態，TAR 狀態表示 TWM 及 TWS 都不驅動 SDA 訊號，此時 SDA 訊號應處於 tri-state，但 TWM 對 SDA 訊號有接一 pull-up resistor，故當 SDA 訊號呈現 tri-state 時會被 pull-up resistor 給設定為 1。因此接下來 TWS 可開始驅動 SDA 訊號。

T8：接著，TWS 擁有 SDA 訊號的控制權。TWS 將 SDA 設為 1，並進行資料讀取的動作。從這時候開始，便改由 TWS 送訊號，而 TWM 負責接收，TWM(testbench)會在每個 SCL **負緣**將 SDA 送出的訊號進行取樣。TWS 可將 SDA 設為 1 持續數個週期，直到資料讀取完成。

T9：當 TWS 進行資料讀取完成後，將 SDA 設定為 0。表示準備開始輸出所讀取的資料。

T10~T11：TWS 將讀取完成的資料由 LSB 開始利用 SDA 依序送出。TWM(testbench)會在每個 SCL **負緣**將 SDA 送出的訊號進行取樣，如上圖紅色虛線處。

T12：資料已經傳送完成，TWS 將 SDA 設定為 1 並準備將 SDA 變成 TAR 狀態。

T13：接下來 SDA 再次變為 TAR 狀態。此時 TWM 及 TWS 都不驅動 SDA 訊號。

T14：接下來 TWM 再次擁有 SDA 訊號的控制權，並將 SDA 設為 1，此時 TWP 系統為閒置的狀態。等同 T1 時間。

3. 評分標準

主辦單位的評分人員將依照參賽者提供之系統時脈進行 RTL simulation 或 Gate-level simulation，以驗證設計正確性。各參賽隊伍應於參賽者定義的系統時脈下，確保輸出結果無設置與保持時間(setup/hold time)的問題，並完全符合主辦單位所提供的標準設計結果。

✧ **評分項目一：依"模擬時間"(Time)長短評分**

各參賽隊伍執行RTL Simulation及Pre-layout Gate-level 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 = 148298 NS 做評分。

```
----------------------------------------------------

Congratulations! All data have been generated successfully!

-------------------------PASS-----------------------

Simulation complete via $finish(1) at time 148298 NS + 0
./testbench.v:215 $finish;
```

✧ **評分項目二：依"面積"(Area)大小評分**

各參賽隊伍完成電路合成後，Cell Area 可利用以下指令產生而得知

以 DC 產生 QoR report 的指令 ： report_qor > TPA.qor

以 RC 產生 QoR report 的指令 ： report qor > TPA.qor

評分人員會以此 TPA.qor 如下面範例，紀錄成 Area = 218075 (以小數點後做四捨五入)做評分。

```
Area
--------------------------------------
Combinational Area:     66483.764989
Noncombinational Area:
                       151591.403398
Buf/Inv Area:            9919.605447
Total Buffer Area:           9715.92
Total Inverter Area:          203.69
Macro/Black Box Area:       0.000000
Net Area:              1834056.160553
--------------------------------------
Cell Area:              218075.168386
Design Area:           2052131.328939
```

✧ **評分方法及標準**

評分方式會依設計完成程度，分成 A、B、C 三種等級，排名順序為 **A > B > C**。主辦單位將根據設計內容的完成度給予記分。

1. **A 等級**： RTL 與 Gate-level simulation 結果完全正確。

   此等級之成績計算方式如下：

   Score = Time x Area

   註: Time 以 **Gate-level simulation** 時間計算。

   註: 本等級中，**Score 越小者**為同級名次越好!

2. **B 等級** ： RTL simulation 結果完全正確，但 Gate-level simulation 結果出現錯誤。

   在 clock cycle 設定為 10ns 的情況下，此等級之成績計算方式如下：

   Score = Time

   註: 本等級中，**Score 越小者**為同級名次越好!

3. C 等級: RTL simulation 結果出現錯誤，則依測試項目 Stage 通過數量評分，通過越多 Stage 者越高分。

附錄

附錄 A 中說明本次競賽之軟體環境；
附錄 B 為主辦單位所提供各參賽者的設計檔說明；
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 則為設計檔整理步驟說明；
附錄 E 則為設計資料庫說明。

附錄 A 軟體環境

競賽所提供的設計軟體環境如下表二。驗證評分時，係以所列軟體作為驗證依據。

1. 表二、設計軟體環境

Vendor	Tool	Executable
Cadence	Virtuoso *1	icfb
	Composer	icfb
	NC-Verilog	ncverilog
	SOC Encounter	encounter
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	vsim
Utility	vi	vi, vim
	gedit	gedit
	nedit	nedit
	pdf reader	acroread
	calculate	gnome-calculator, bc -l
	gcc	gcc
	Matlab	matlab

EDA 軟體所需使用的 license 皆已設定完成，不須額外設定
*1 表示該軟體限定使用 1 套 license
*3 表示該軟體限定使用 3 套 license

附錄 B 設計檔案說明

1. 下表三.為主辦單位所提供各參賽者的設計檔案

表三、設計檔

檔名	說明
testbench.v	測試樣本檔。此測試樣本檔定義了時脈週期與測試樣本之輸入及預期輸出信號。
TPA.v	參賽者所使用的設計檔，已包含系統輸/出入埠之宣告
Pattern_sti.dat	測試樣本檔案
Pattern_exp.dat	比對樣本檔案
TPA.sdc	Design Compiler 電路合成規範檔。 **本規範檔除了 cycle 可修改外，其餘 constraints 皆不可修改。**
tsmc13_neg.v	Gate-level simulation 所需要之 cell library file
synopsys_dc.setup	Design Compiler 初始設定範例檔案

2. 請使用 TPA.v，進行本題電路之設計。其 Verilog 模組名稱、輸出/入埠宣告如下所示：

```
module TPA(clk, reset_n,
           SCL, SDA,
           cfg_req, cfg_rdy, cfg_cmd, cfg_addr, cfg_wdata, cfg_rdata);
input           clk;
input           reset_n;
// Two-Wire Protocol slave interface
input           SCL;
inout           SDA;

// Register Protocal Master interface
input           cfg_req;
output          cfg_rdy;
input           cfg_cmd;
input   [7:0]   cfg_addr;
input   [15:0]  cfg_wdata;
output  [15:0]  cfg_rdata;

reg     [15:0] Register_Spaces  [0:255];

// ===== Coding your RTL below here ================================



endmodule
```

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

```
`define   sdf_file      "./TPA_syn.sdf"

`ifdef   SDF
   initial   $sdf_annotate(`sdf_file, u_TPA);
`endif
```

註:

1. SDF 檔案，請自行修改 SDF 實際檔案名稱及路徑後再模擬。
2. 在 Testbench 中，本題提供 `ifdef SDF 的描述，目的是讓 Testbench 可做為 RTL 模擬及合成後模擬皆可使用。當參賽者在合成後模擬時，請務必多加上一個參數 +define+SDF ，方可順利模擬。
3. 合成後的 Gate-Level Simulation 模擬指令範例如下：
   ncverilog Testbench.v TPA_syn.v –v tsmc13_neg.v +define+SDF

4. 比賽共提供一組測試樣本，參賽者可依下面範例來進行模擬:
   主辦單位於評分時，將可能額外再使用其他組測試樣本進行評分工作。
   RTL Simulation 時使用指令如下 ：
   - 使用 ncverilog 模擬指令範例如下：
     ncverilog Testbench.v TPA.v
   - 使用 modelsim 模擬，則是在 compiler verilog 時，使用下面指令：
     vlog Testbench.v

   Gate-Level Simulation 時使用指令如下 ：
   - 使用 ncverilog 模擬指令範例如下：
     ncverilog Testbench.v TPA_syn.v –v tsmc13_neg.v +define+SDF
     define 中加上 SDF 可讓測試程式引入 gate level netlist 的 sdf 檔案資訊。

   - NC-Verilog 使用者若想要輸出 FSDB 格式波形檔案，
     可自行再加入 *+define+FSDB* ***+access+r***
     modelsim 使用者，請直接使用內建波形來進行除錯。

附錄 C 評分用檔案

評分所需檔案可分為三部份：(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，請務必將使用到的各 module 檔放進來，以免評審進行評分時，無法進行編譯；(2)Gate-level design，即由合成軟體所產生的 Gate-level netlist，以及對應的 SDF 檔。

表四、評分用檔案

RTL category		
Design Stage	File	Description
N/A	report.xxx	design report
RTL Simulation	*.v or *.vhd	Verilog (or VHDL) synthesizable RTL code
Gate-Level category		
Design Stage	File	Description
Pre-layout Gate-level Simulation	*_syn.v	Verilog gate-level netlist generated by Synopsys Design Compiler
	*_syn.sdf	SDF timing information generated by Synopsys Design Compiler
	*_syn.ddc	design database generated by Synopsys Design Compiler

附錄 D 檔案整理步驟

當所有文件準備齊全如表四所列，請按照以下步驟指令，提交相關設計檔案，將所有檔案複製到同一資料夾下，步驟如下:

1. 在自己的 home directory 建立一個新目錄，名稱叫做“ result ”， 範例如下:
   > mkdir ~/result
2. 將附錄 C 中要求的檔案全部複製到 result 這個目錄。範例如下:
   > cp TPA.v ~/result/
   > cp TPA_syn.v ~/result/
   其他檔案依此類推……
3. 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/icc2018/CBDK_IC_Contest_v2.1**

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

Design Report Form

登入帳號(login-id)		
RTL category		
Design Stage	*Description*	*File Name*
RTL Simulation	使用之 HDL 名稱 (例如：Verilog、VHDL)	
RTL Simulation	RTL 檔案名稱 (RTL Netlist file name)	
Gate-Level category		
Design Stage	*Description*	*File Name*
Pre-layout Gate-level Simulation	Gate-Level 檔案名稱 (Gate-Level Netlist file name) (*_syn.v)	
	Pre-layout sdf 檔案名稱 (*_syn.sdf)	
	Design database 檔案名稱 (*_syn.ddc)	
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		