

F 組-大學類可程式邏輯設計

初賽時間為 3 月 24 日(三) **8:30~20:30**，初賽當日請密切注意 IC 競賽網頁與 CIC 網頁，所有最新訊息將於網頁上公告。請參賽隊伍於**早上 10 點半前**完成下列步驟進行初賽登錄，主辦單位將依完成此步驟之隊伍數決定各組最後得獎名額，請務必完成登錄動作，以免影響您的權益。

※請參賽者將作品在**今天 20:30** 前上傳至 FTP。

請將您的隊伍參賽資料E-mail至 icdesign_d@yuntech.edu.tw

- 信件內容格式如下：
 - 參賽組別：F 組-大學類可程式邏輯設計**
 - 參賽編號：(例：60001)**
 - 參賽姓名：張三、李四**
- 信件標題請標示為「**IC 設計競賽初賽資料登錄**」

※初賽之注意事項請參閱競賽手冊

2010 IC Design Contest Preliminary

大學部可程式邏輯設計

1. 問題描述

請完成一序列傳輸/接收電路(Serial Interface transmitter/receiver)(如圖一)。參賽者需要完成序列埠(sd)兩端 S1 及 S2 的電路設計。本序列傳輸/接收電路負責將記憶體 RB1(Register Bank1)的資料經由序列埠上傳到記憶體 RB2(Register Bank2)。

本傳輸/接收電路中，S1 有三只信號輸入(clk、rst、RB1_Q)、五只信號輸出(RB1_RW、RB1_A、RB1_D、sen、sd)。S2 有五只信號輸入(clk、rst、RB2_Q、sen、sd)、四只訊號輸出(S2_done、RB2_RW、RB2_A、RB2_D)。相關信號的說明，請參考表一和表二。

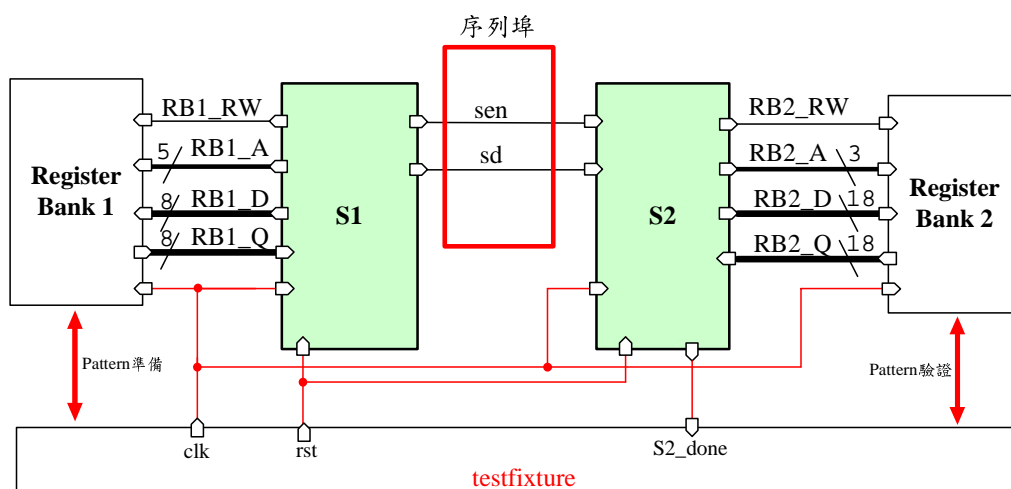
本試題有使用到記憶體模型(memory simulation model)，詳細的記憶體規格描述在 2.2.1。本組初賽僅檢查 RTL simulation 結果是否正確與電路是否可合成，請參賽同學務必配合題目卷之要求進行作答。

每個參賽隊伍必須根據下一節所給的設計規格完成設計。參賽隊伍可藉由 CIC 所提供的測試樣本檔(testbench) tb.v 來檢查設計是否有達到要求，CIC 所提供的相關檔案請參考附錄 B。

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 D 中所列的要求，上傳評分所需要的檔案。

2. 設計規格

2.1 系統方塊圖



圖一、系統方塊圖

2.2 輸入輸出訊號和記憶體描述

表一、S1 輸入/輸出信號

信號名稱	輸入/輸出	位元寬度	說明
<i>clk</i>	input	1	系統提供的時脈信號。
<i>rst</i>	input	1	高位準非同步(active high asynchronous)之系統重置信號。 說明：此信號於系統啟動時送出。
<i>RB1_RW</i>	output	1	Register Bank1 讀寫控制信號。 說明：當此信號為 high 時是對 Register Bank1 進行讀取(read)；low 則是進行對 Register Bank1 寫入(write)。
<i>RB1_A</i>	output	5	Register Bank1 五位元位址信號。
<i>RB1_D</i>	output	8	Register Bank1 八位元資料輸入埠。
<i>RB1_Q</i>	input	8	Register Bank1 八位元資料輸出埠。
<i>sen</i>	output	1	致能信號。 說明：當此信號為 low 時，sd 傳輸的資料被認為是有效的。
<i>sd</i>	output	1	序列傳輸中的一位元資料輸出埠。資料在 clk 負緣的時候觸發。

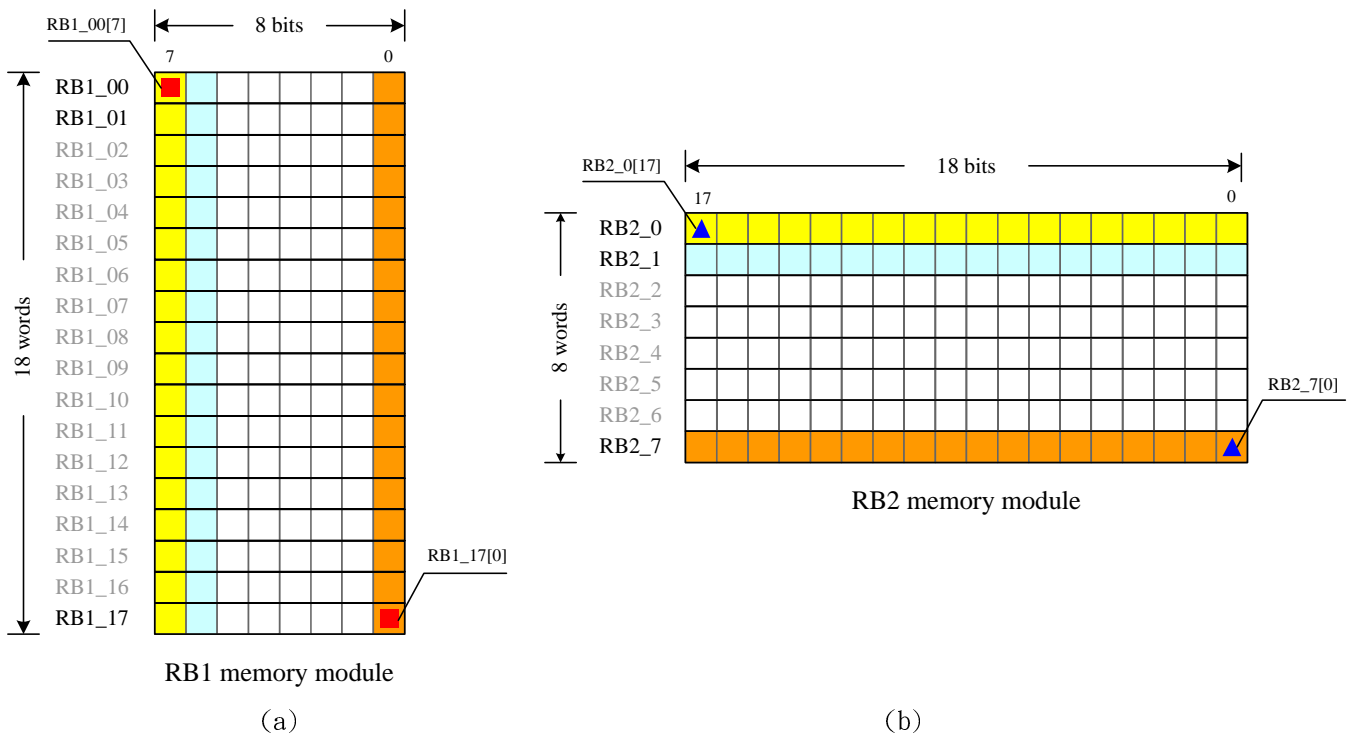
表二、S2 輸入/輸出信號

信號名稱	輸入/輸出	位元寬度	說明
<i>clk</i>	input	1	系統提供的時脈信號。
<i>rst</i>	input	1	高位準非同步(active high asynchronous)之系統重置信號。 說明：此信號於系統啟動時送出。
<i>S2_done</i>	output	1	當 S2 寫入 RB2 完成的時後，將 S2_done 設為 1 表示完成。
<i>RB2_RW</i>	output	1	Register Bank2 讀寫控制信號。 說明：當此信號為 high 時是對 Register Bank2 進行讀取(read)；low 則是進行對 Register Bank2 寫入(write)。
<i>RB2_A</i>	output	3	Register Bank2 三位元位址信號。
<i>RB2_D</i>	output	18	Register Bank2 十八位元資料輸入埠。
<i>RB2_Q</i>	input	18	Register Bank2 十八位元資料輸出埠。
<i>sen</i>	input	1	致能信號。 說明：當此信號為 low 時，sd 傳輸的資料被認為是有效的。
<i>sd</i>	input	1	序列傳輸中的一位元資料輸入埠。S2 於 clk 正緣時對該訊號取樣(sampling)。

2.2.1 Register Bank 1 (RB1)和 Register Bank2 (RB2)規格描述

本次試題使用到兩個記憶體模組 Register Bank 1 (以下稱為 RB1)及 Register Bank 2 (以下稱為 RB2)，兩者都是單埠(single port)的 Register File 格式。RB1 的記憶體寬度是 8 位元，而記憶體深度是 18 個 word (請參考圖二(a))。如果要描述 RB1 中第一個 word 中的話，會用 RB1_00 表示；若要描述該 word 的 MSB 的話，會用 RB1_00[7]表示。同理，如果要描述 RB1 中最後一個 word 中的話，會用 RB1_17 表示；而最後一個 word 的 LSB 會用 RB1_17[0]表示。

RB2 的記憶體寬度是 18 位元，而記憶體深度是 8 個 word (請參考圖二(b))。同上面說明，RB2 中第一個 word 中的 MSB 的話會用 RB2_0[17]表示，最後一個 word 的 LSB 會用 RB2_7[0]表示。



圖二、RB1 的示意圖(a)和 RB2 的示意圖(b)

2.3 系統功能描述

本序列/傳輸接收電路具有資料上傳(將 RB1 資料搬移至 RB2 中)的功能：

當發生系統重置(rst)時，S1 將 RB1 中的資料，轉換成為帶有記憶體位址和資料的封包(詳細的封包格式請參考 2.3.1)，透過序列埠傳送到 S2。S2 收到訊號之後，根據封包內的記憶體位址將資料存放到 RB2 中。即本系統共實現 S1 及 S2 兩塊電路；S1 為一個模組，S2 為另一個模組。

整個系統工作方式如下：

- I. 首先測試樣本(testbench)會將 RB1 中填滿資料，並且把重置訊號 rst 設為 0
- II. Testfixture 將 rst 設為 1，對系統進行 RESET。此時 S1 的輸出 RB1_RW 及 sen 為 1，其餘 (RB1_A、RB1_D、sd)輸出為 0；此時 S2 的輸出中 RB2_RW 為 1，其餘(RB1_A、RB1_D、

S2_done)輸出為 0

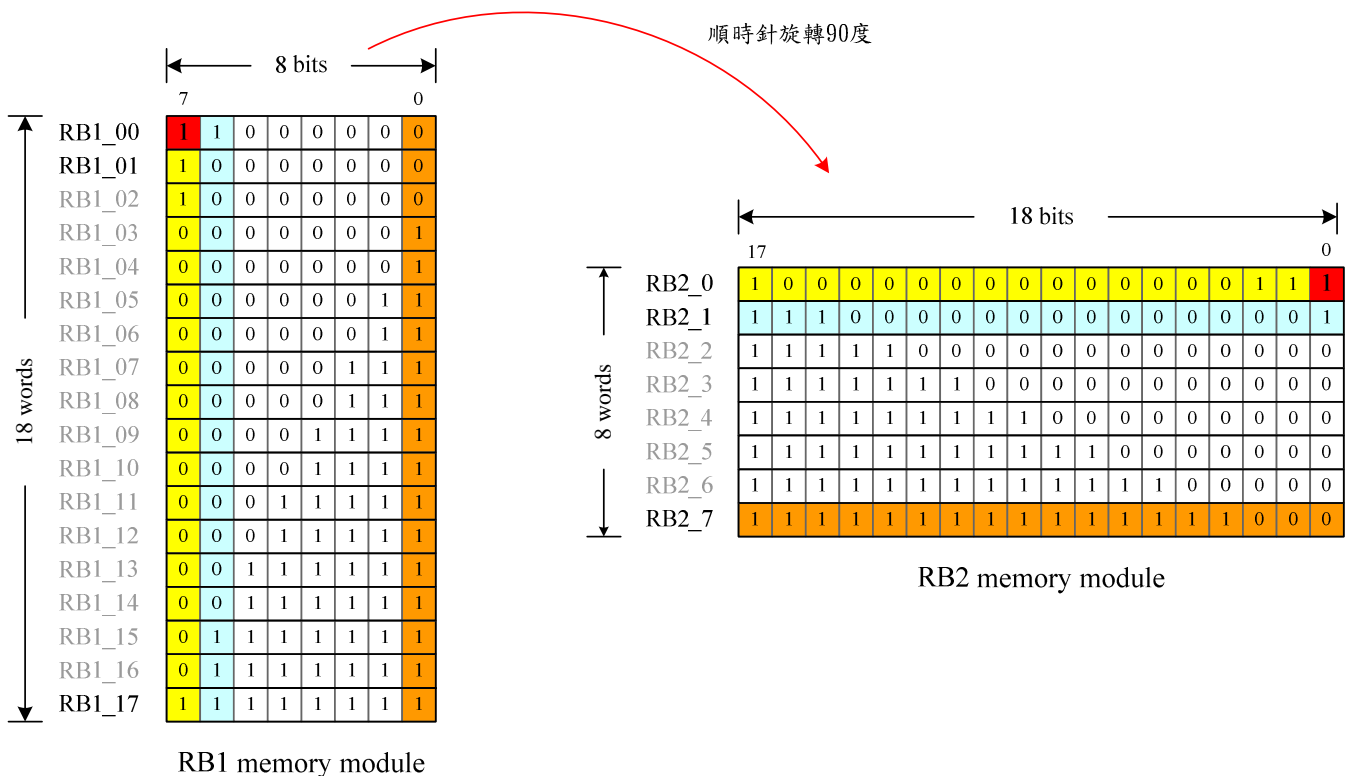
- III. 當 S1 及 S2 電路完成 RESET 後，測試樣本將 rst 設為 0，此時 S1 開始將 RB1 的資料轉換成為序列封包的格式傳送，每次傳送是將 RB1_00 到 RB1_17 的第 n 個 bit (n 是 MSB 到 LSB)包裝成一個封包送出 (詳見 2.3.1)；S2 自 RESET 後即等著接收封包，一旦接收到封包並根據封包內所存的位址，將資料儲存在 RB2 對應的位址中。

S1 是在 clk 負緣觸發的時候透過 sd 送出序列資料，而 S2 會在 clk 正緣的時候從 sd 接收資料(詳細 clk, sen 和 sd 的相關時序關係描述在 2.4)

- IV. 當 RB2 所有資料都儲存完成之後(共八個封包)，S2 將 S2_done 設為 1 表示已完成上傳的動作。

RB1 和 RB2 資料對應關係：

RB1 及 RB2 記憶體模組是用來儲存傳送及接收的資料。當 RB1 中的資料依照系統功能描述的要求，完成上傳到 RB2 的動作後，RB2 中的資料分布方式將有如把 RB1 整個「順時針旋轉 90 度」(如圖三所示)。也就是說原本資料是用 18x8 的方式儲存在 RB1 中，上傳後在 RB2 中將以 8x18 的方式儲存。其中 RB1_17[7]這一 bit 資料將被上傳至 RB2_0[17]，RB1_16[7]這一 bit 資料將被上傳至 RB2_0[16]，...同理，RB1_17[6]這一 bit 資料將被上傳到 RB2_1[17]，RB1_16[6]這一 bit 資料將被上傳至 RB2_1[16]的位置，...其餘類推。本試題並不評比電路所使用的硬體資源，學生可以根據設計考量使用足夠大的緩衝空間。請注意，圖三中記憶體的資料內容僅於此處作說明使用，與本次競賽實際使用之測試樣本內容並不一致，請勿誤解。



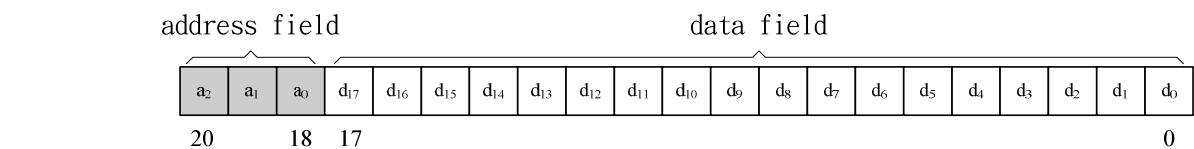
圖三、RB1 資料上傳到 RB2 範例

2.3.1 封包格式

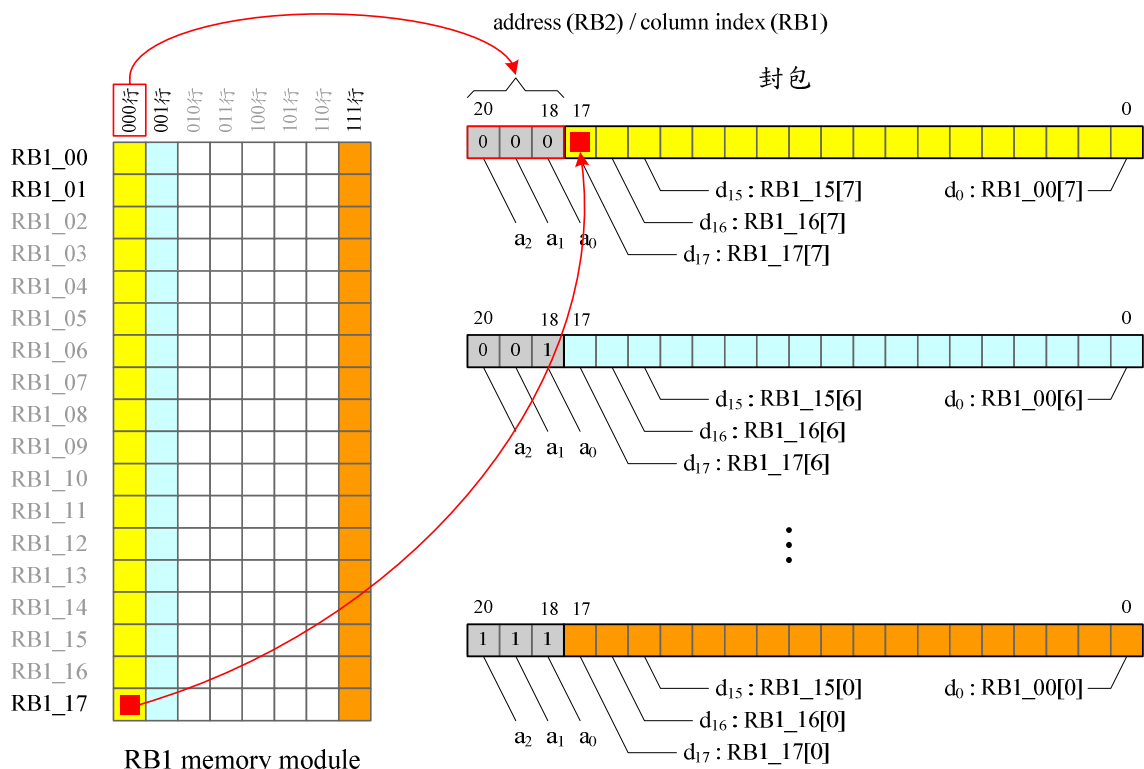
如圖四(a)所示，資料封包由 3 bit 的位址欄(address field)及 18 bit 的資料欄(data field)兩部份組成。S1 電路將整塊 RB1 記憶體中的資料以「行」(column)的方式切分成八個集合，從 MSB 到 LSB 依序編號(index)為 000、001、010、...、111。因為編號 000 的資料集合經過傳送後將儲存在 RB2 的 000 位址處，編號 001 的資料集合經傳送後將儲存在 RB2 的 001 位址處，其餘類推；由此可知該編號恰好代表此封包內資料預定儲放在 RB2 的位址(address)。封包的第一部份(位址欄)即由此 3 bit 編號組成。此 3 bit 於傳送時將由 MSB 開始傳送。

而 RB1 的資料切分出的每一個集合，即構成各封包的資料欄(即封包的第二部份)。以編號 000 的資料部份為例，其 18 個 bit 的資料欄的第一個 bit 是編號 000 的資料集合與 RB1_17 的交集，接著第 2 個 bit 是編號 000 的資料集合與 RB1_16 的交集，...，第 18 個 bit (最後一個 bit)則是編號 000 的資料集合與 RB1_00 的交集(即由 RB1 的高位址往低位址方向依序排列)，如圖四(b)所示。

資料進行傳送時，S1 將由編號 000 的封包開始依序(sequential)傳送到 S2 側。



(a)封包格式

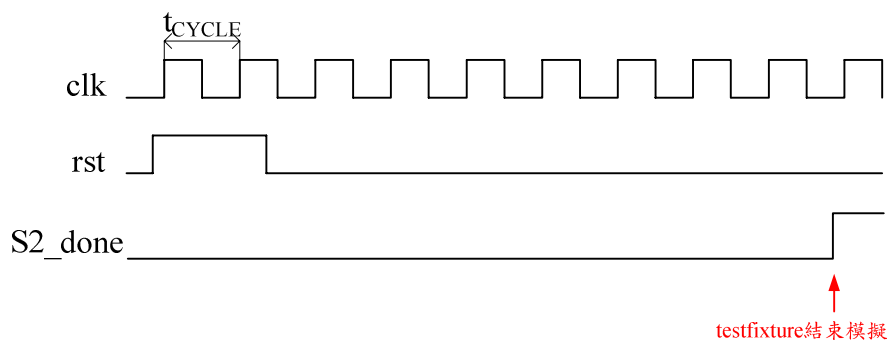


(b)封包的組成

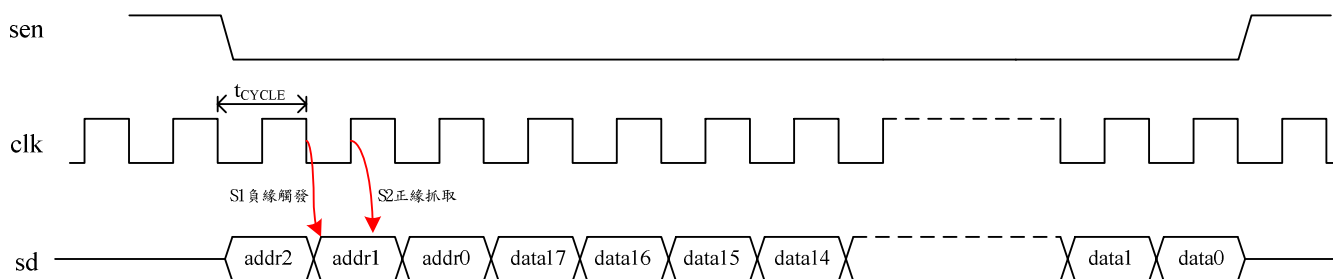
圖四、RB1 資料與封包的關係圖

2.3.2 時序規格圖

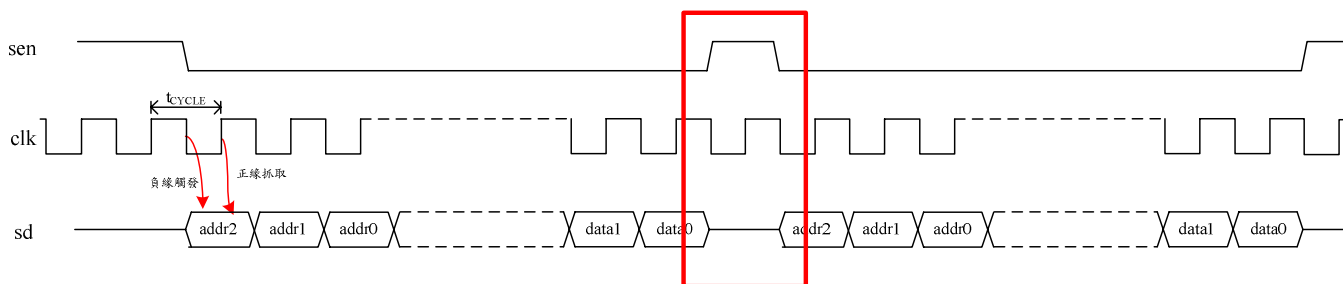
圖五為本系統中 clk、rst 及 S2_done 訊號之時序圖，本電路工作時脈 clk 之週期 t_{CYCLE} 固定為 100ns。rst 會舉為 high 至少維持一個週期以上。測試樣本會在時脈正緣的時候檢查 S2_done 是否為 1；當 S2_done 為 1 的時候，測試樣本會檢查 RB2 中的資料是否正確，然後結束模擬。圖六是 S1 傳送序列資料到 S2 的時候，序列傳輸埠 sd 及 sen 的波形圖，可以看到 S1 在負緣觸發的時候送出資料，而整個封包有效的情況是在 sen 維持 low 的時候，之後在由 S2 在 clk 正緣的時候抓取資料。而在傳輸過程中，封包與封包間至少要間隔一個週期(請參考圖七說明)。



圖五、系統時序圖



圖六、S1 上傳 S2 操作關係圖

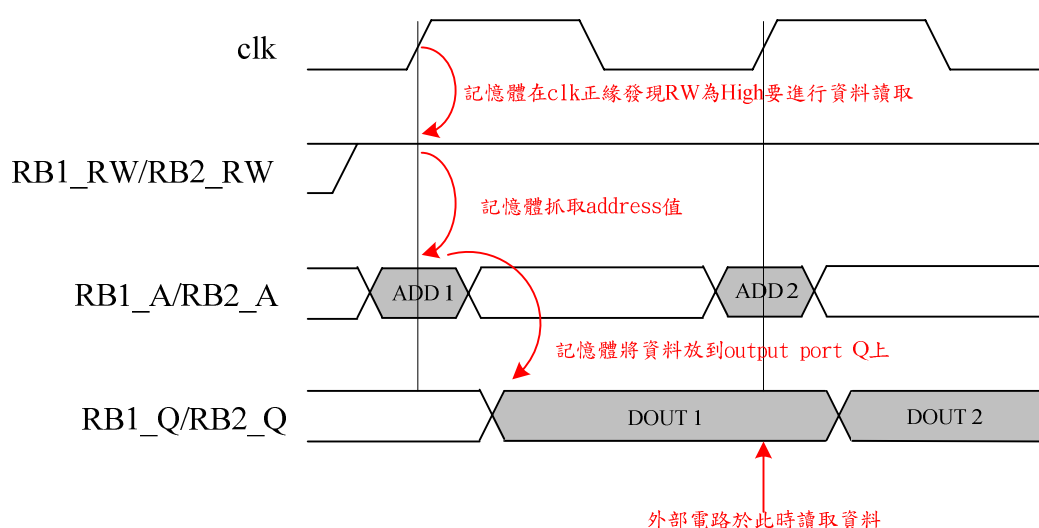


圖七、連續封包的上傳關係圖

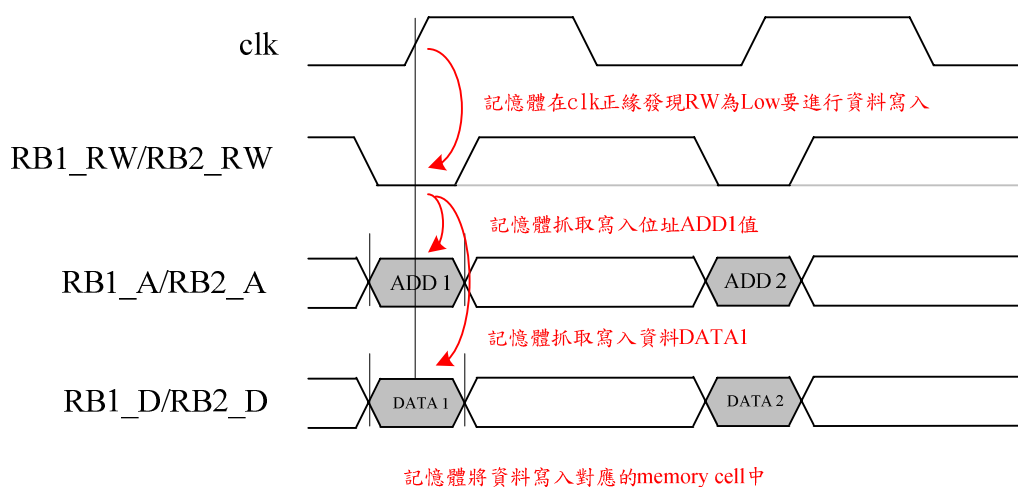
2.4 RB1 和 RB2 之時序規格

RB1 和 RB2 的讀取與寫入動作主要是以 RB1_RW/RB2_RW 這個信號來做控制。圖八為對記憶體進行資料讀取的波形時序圖，記憶體在 clk 的正緣確認到 RB1_RW/RB2_RW 為 High 時，會將此時的 RB1_A/RB2_A 位址匯流排上的 address 所對應到的資料，放到資料讀取匯流排(Q) RB1_Q/RB2_Q 上，以便在下一個 clk 正緣供電路讀取。圖九為對記憶體進行資料寫入的波形時序圖，記憶體在 clk 的正緣確認到 RB1_RW/RB2_RW 為 Low 時，會將此時的資料寫入匯流排(D) RB1_D/RB2_D 上的資料，寫到此時 RB1_A/RB2_A 位址匯流排所對應的記憶體單元上。**特別注意**若沒有要進行寫入時，請將 RB1_RW/RB2_RW 保持為 high，維持在讀取狀態。

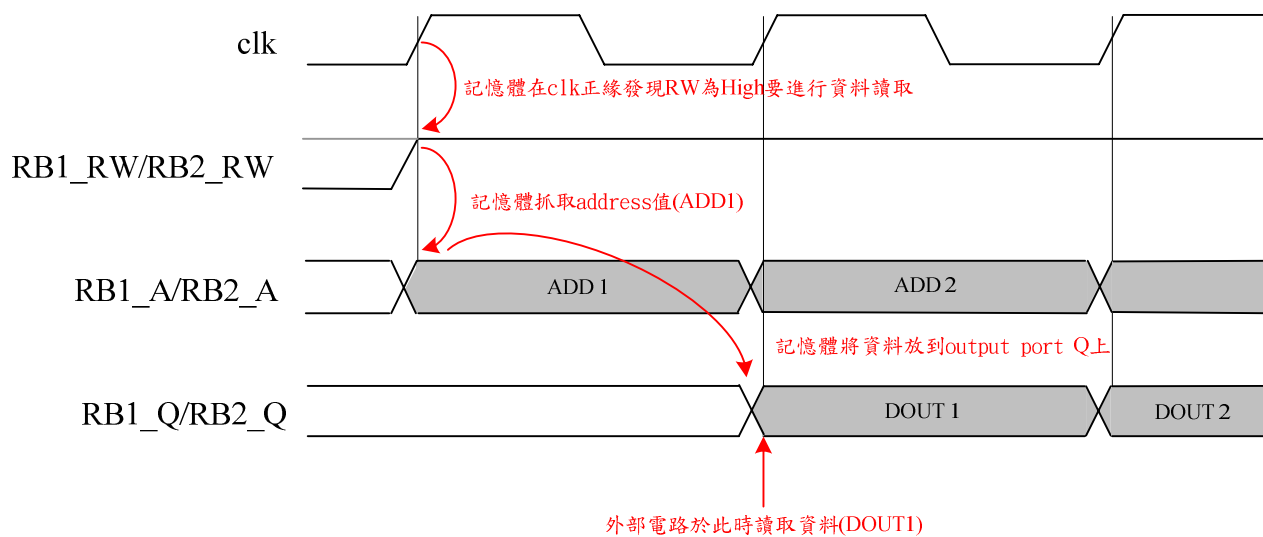
本次競賽以 RTL simulation 結果作評分之依據，RB1 及 RB2 於 RTL simulation 的讀寫時序圖請參考圖十及圖十一。



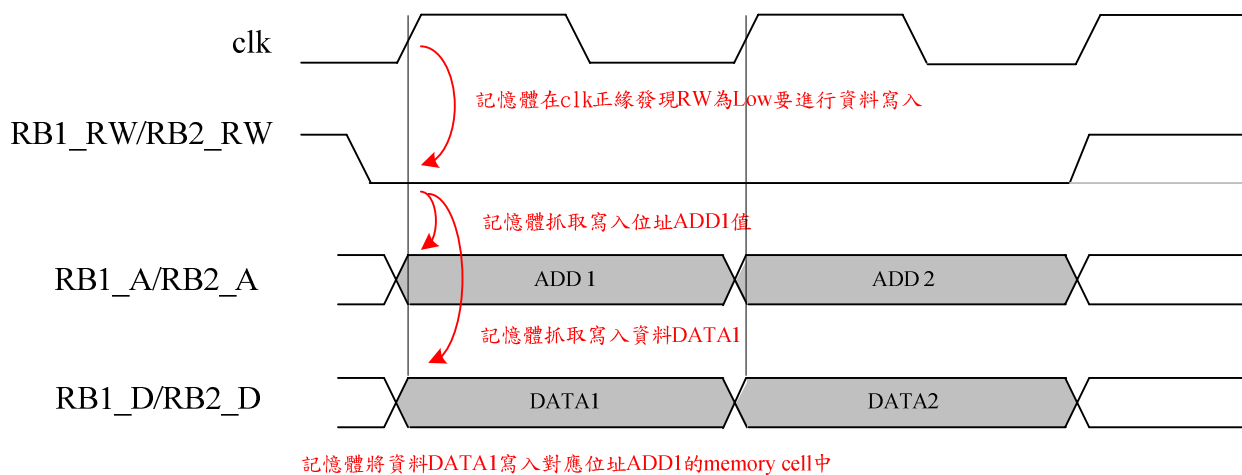
圖八、RB1/RB2 資料讀取波形



圖九、RB1/RB2 資料寫入波形



圖十、RB1/RB2 資料讀取波形(RTL simulation)



圖十一、RB1/RB2 資料寫入波形(RTL simulation)

2.5 FPGA 元件指定

本次 FPGA 組初賽主要由 Mentor Graphics 公司 Modelsim 進行 RTL 模擬來進行評分，唯各參賽者所設計之 S1 及 S2 模組必須可分別在 FPGA 開發工具進行電路合成。請各參賽者依使用 Altera 或是 Xilinx FPGA 開發工具之不同，選擇如下的元件及對應之 FPGA 發展工具進行設計編譯。請參賽者注意，**本試題第三節所述的評分標準中，關於元件資源使用率(usage ratio)及設計效能，或選用不同之 FPGA 開發工具，並不會影響本競賽初賽的評分!! 且各位參賽者需在 report file 中填寫所使用之合成器為 Quartus II 或是 ISE。**本次競賽所使用之 FPGA 開發環境軟體版本以附錄 A 中所指定為標準。

表三、Altera 組元件指定

Altera	
Target Family	Cyclone II
Target Device	EP2C35
Target Package	FBGA
Pin count	672
Target Speed	6

表四、Xilinx 組元件指定

Xilinx	
Target Family	Spartan 3
Target Device	XC3S1500
Target Package	FG (FBGA)
Pin count	676
Target Speed	-4

3. 評分標準

主辦單位的評分人員將依照參賽者提供之設計電路進行 RTL simulation，以驗證設計正確性，並且依據設計檔上傳至 CIC FTP 檔案伺服器(請參閱附錄 D)的時間來進行排名。各參賽隊伍應於規定的系統時脈下，確保電路可合成且滿足功能驗證(functional verification)，並完全通過 CIC 所提供的測試樣本為準。

CIC 將本試題依 RTL simulation 及電路合成結果區分為下面 3 個等級，以作為功能完成度之評分；若為同一等級則以檔案上傳時間(即作答時間的長短)來評分：

1. A 等級：S1 及 S2 電路的 RTL simulation 均通過測試樣本一及測試樣本二 (tb1 & tb2)之功能驗證，且 S1、S2 均可合成
2. B 等級：S1 電路的 RTL simulation 通過測試樣本一及測試樣本二(tb1 & tb2)之功能驗證，S2 電路功能驗證失敗，且 S1 電路可合成
3. C 等級：S1 及 S2 的 RTL simulation 結果均 fail

請注意，我們將以各參賽隊伍的設計結果正確為前提，並以最後上傳檔案的時間為依據。一旦設計經評審驗證後，完成同一等級者，上傳時間越早，其所得到的分數就越高。建議每完成一個等級就先將設計檔案內容上傳，主辦單位將根據設計內容的完成度給予記分。審查成績將另擇期通知。

附錄

在附錄 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，亦即參賽者必須回傳至 CIC 的檔案資料；附錄 D 則為設計檔上傳步驟說明。

附錄 A 軟體環境

競賽主辦單位所提供的設計軟體與版本如下表(表五)。驗證評分時，係以所列軟體及版本作為驗證依據。

表五、設計軟體版本

Functionality	Corresponding EDA tools
Design Entry	VHDL/Verilog
FPGA Synthesizer	Mentor Graphics Precision RTL v2005c.99
FPGA Simulator	Mentor Graphics ModelSim 6.3a
FPGA tools	Altera Quartus II v9.0 or Xilinx ISE v10.1i

其中 ModelSim、Precision RTL 為輔助性工具，並不強制各參賽單位使用。各隊伍可以利用上述 EDA 軟體工具來檢查自己設計的正确性及設計是否可合成，也可直接使用 FPGA 廠商的發展軟體(如 Altera Quartus II 或是 Xilinx ISE)中內建的編輯功能來完成這些動作，這些並不會影響評分的進行(請參考本試題卷第 3 節之評分標準說明)。**主辦單位是以(1)ModelSim 模擬結果及(2)電路是否可合成作為評分的依據。**

如參賽手冊中刊載主辦單位，採用之 ModelSim 為 CIC 提供學術界申請使用的 Mentor Graphics 公司發行之正式版，有支援 Verilog/VHDL 混合模擬的功能，若參賽者使用自行由 Altera/Xilinx 公司下載之 OEM 版 ModelSim 而無法順利利用主辦單位提供的 verilog testbench 進行模擬者，不在主辦單位負責範圍內。

附錄 B 設計與測試樣本檔案說明

1. 以下表六為主辦單位所提供各參賽者的設計檔案

表六、設計檔

檔名	說明
tb.v	測試樣本檔(testbench)。此測試樣本檔定義了時脈週期與測試樣本之輸入信號，module 名稱為 tb
S1.v (S1.vhd)	參賽者 S1 的設計檔範本，已包含系統輸/出入埠之宣告
S2.v (S2.vhd)	參賽者 S2 的設計檔範本，已包含系統輸/出入埠之宣告
RB1.v	Register Bank 1 模擬檔 (simulation model)
RB2.v	Register Bank 1 模擬檔 (simulation model)
tb1_RB1_ini.dat	測試樣本一之 RB1 輸入檔案 (initializing file)
tb1_RB2_ref.dat	測試樣本一之 RB2 比對檔 (reference file)
tb2_RB1_ini.dat	測試樣本二之 RB1 輸入檔案 (initializing file)
tb2_RB2_ref.dat	測試樣本二之 RB2 比對檔 (reference file)
report_000.txt	結果報告範本

2. 請使用 *S1.v(.vhd)/S2.v(.vhd)*，進行序列傳輸接收電路之設計。其模組名稱、輸出/入埠宣告如下列範本所示：

```
// Verilog user: S1.v
module S1(clk, rst, RB1_RW, RB1_A, RB1_D, RB1_Q, sen, sd);
    input clk, rst;
    output RB1_RW;
    output [4:0] RB1_A;
    output [7:0] RB1_D;
    input [7:0] RB1_Q;
    output sen, sd;
endmodule
```

// Verilog user: S2.v

```
module S2(clk, rst, S2_done, RB2_RW, RB2_A, RB2_D, RB2_Q, sen, sd);
    input clk, rst;
    output S2_done, RB2_RW;
    output [2:0] RB2_A;
    output [17:0] RB2_D;
    input [17:0] RB2_Q;
    input sen, sd;

endmodule
```

-- VHDL user: S1.vhd

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity S1 is Port (
    clk      : in    STD_LOGIC;
    rst      : in    STD_LOGIC;
    RB1_RW   : out   STD_LOGIC;
    RB1_A    : out   STD_LOGIC_VECTOR (4 downto 0);
    RB1_D    : out   STD_LOGIC_VECTOR (7 downto 0);
    RB1_Q    : in    STD_LOGIC_VECTOR (7 downto 0);
    sen      : out   STD_LOGIC;
    sd       : out   STD_LOGIC);
end S1;

architecture S1_arc of S1 is

begin

end S1_arc;
```

```

-- VHDL user: S2.vhd
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity S2 is Port (
clk : in  STD_LOGIC;
    rst      : in  STD_LOGIC;
    S2_done  : out  STD_LOGIC;
    RB2_RW   : out  STD_LOGIC;
    RB2_A    : out  STD_LOGIC_VECTOR (2 downto 0);
    RB2_D    : out  STD_LOGIC_VECTOR (17 downto 0);
    RB2_Q    : in   STD_LOGIC_VECTOR (17 downto 0);
    sen      : in   STD_LOGIC;
    sd       : in   STD_LOGIC);
end S2;

architecture S2_arc of S2 is

begin

end S2_arc;

```

3. 比賽共提供兩組測試樣本。請自行修改 tb.v 內容，來決定模擬時使用那一組測試樣本。tb.v 內容第一行：

```
`define tb1
```

表示使用第一組測試樣本。若要使用第二組測試樣本，請將上面這行修改為：

```
`define tb2
```

4. 比賽共提供兩組測試樣本，**參賽者可依下面範例來進行模擬：**

- FPGA 組使用 modelsim 進行模擬，在 compile verilog 時，使用下面指令：

vlog verilog_filename.v

- FPGA 組使用 modelsim 進行模擬，在 compile VHDL 時使用下面指令：

vcom VHDL_filename.vhd

- 關於模擬時使用的一些記憶體模型，因已經以 include 方式加在 tb.v 裏，所以不需加在模擬指令裏。即對 tb.v 進行 compile 時，RB1.v 及 RB2.v 即自動跟著一起 compile。
- 若 RTL 模擬時，為了確保足夠的可視範圍(visibility)，可在載入時加上「-novopt」參數以關閉最佳化功能，其參考指令語法如下(work 為工作 library 之名稱，tb 為測試樣本的 module name):

vsim -novopt work.tb

- 提醒 VHDL 的使用者，如同參賽手冊所載明：比賽時使用 CIC 提供之 Mentor Graphics 公司的正式版 ModelSim。該版本之 ModelSim 有支援 Mix-language (即整個系統中混用 Verilog 及 VHDL 作設計)的模擬功能，為保持測試樣本之一致性，故未再提供 VHDL 版的測試樣本與 memory simulation model，但並無妨礙。參賽者只要掌握「對設計中的.v 檔用 vlog 作 compile」、「對設計中的.vhd 檔用 vcom 作 compile」、「最後在載入模擬時，不分 verilog 或 VHDL，一律用 vsim -novopt work.tb 指令」的基本原則即可，細節請自行參考 ModelSim 使用手冊。**若參賽者未詳閱參賽手冊之說明並遵行之，而使用自行下載之其它版本 ModelSim 軟體(如 Altera 或 Xilinx 公司提供的 OEM 版 ModelSim)，致使比賽時設計環境出問題造成不便，請自行負責。**

附錄 C 評分用檔案

如表七所示，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，無法進行編譯；(2)Post-layout gate-level netlist，即由 FPGA 發展軟體所產生的 Post-layout gate-level netlist，以及對應的 SDF 檔(Altera 組為*.vo、*_v.sdo 檔或*.vho、*_vhd.sdo 檔，Xilinx 組為*_timesim.v、*_timesim.vhd 及*_timesim.sdf 檔)；(3)report file，參賽隊伍必須依照自己的設計內容，撰寫 report_xxx.txt 檔，以方便主辦單位進行評分，report_xxx.txt 的格式如圖十二所示。(report 檔以後三碼序號表示版本，首次繳交為 report_001.txt。若繳交檔案更新版本，則新版的 report 檔檔名為 report_002.txt，依此類推)。**請注意，本次初賽之電路設計並未開放使用 FPGA 廠商之 IP(如 Altera 之 Megafunction Wizard 及 Xilinx 之 Core Generator)，若參賽者之設計中有使用到 Altera 的 Megafunction/Xilinx 的 Co-gen & Architecture IP 的話，將不予計分。**

表七、評分用檔案

<i>Result_xxx directory</i>	
<i>RTL category</i>	
<i>File format</i>	<i>Description</i>
*.v or *.vhd	Verilog (or VHDL) synthesizable RTL code
<i>Post-Layout Gate-Level category</i>	
<i>File format</i>	<i>Description</i>
*.vo, *.vho, *.sdo (*v, *.vhd, *.sdf)	Verilog/VHDL post-layout gate-level netlist generated by FPGA Development Tool, and SDF file
<i>Report category</i>	
<i>File format</i>	<i>Description</i>
Report_xxx.txt	Design report file

舉例來說，假設參賽隊伍採 Verilog 設計語言，並以模組化設計將系統切分為 fc.v、sub1.v 及 sub2.v，經 FPGA 發展軟體(以 Quartus II 為例)合成後所產生的 post-layout gate-level netlist 為 fc.vo，所產生的 SDF 檔為 fc_v.sdo，則參賽隊伍須將 fc.v、sub1.v、sub2.v、fc.vo、fc_v.sdo、report_xxx.txt 等檔案確實放置於同一個 Result_xxx 目錄下。其中 report_xxx.txt 的內容可參考下面圖十二所示：

隊號(Team number): 60001

--- RTL category---

使用之 HDL 語言:

Verilog

使用之合成器:

Quartus II

RTL 檔案名稱(RTL filename):

fc.v, sub1.v, sub2.v

Post-layout gate-level Netlist:

fc.vo, fc_v.sdo

Report:

report_001.txt (此次為第一次上傳)

-----注意事項(annotation)-----

(其餘注意事項依各參賽隊伍的需求填寫，非為必要部份)

fc 電路 RTL 設計檔之 Top module 為 fc，位於 fc.v 內

圖十二、report_000.txt 的範本

附錄 D 檔案上傳

所有包含於如附錄 C 中表格所示的檔案，均需要提交至 CIC。提交的設計檔案，需要經過壓縮於同一個資料夾下，並為*.zip 或*.rar 格式(建議以*.zip 格式為優先)¹。步驟如下：

1. 建立一個 result_xxx 資料夾。其中“xxx”表示繳交版本(即第幾次上傳之編號)。例如“001”表示為第一次上傳；“002”表示為第二度上傳；003 表示為第三度上傳，以此類推...
2. 參考附錄 C 評分用檔案，將所有繳交檔案複製到 result_xxx 資料夾
3. 參賽者需依據設計內容編輯報告檔案“report_xxx.txt”(如圖十二所示)，並變更其檔名。例如，設計的最初(第一次上傳)版本，報告檔名應命名為“report_001.txt”並置於 Result_001 資料夾下。倘若參賽者於上傳設計後，若需再變更設計，則需進行第二次上傳以便更新(update)，此時檔名應改為“report_002.txt”並置於 Result_002 資料夾下，再連同設計檔一併壓縮上傳。亦即 Result_xxx 與 report_xxx.txt 之“xxx”編號需一致。而評審將以**最後上傳的設計檔及報告檔編號進行評分作業**。
4. 確認該提交的檔案均已備妥，即可將 Result_xxx 資料夾進行壓縮成*.zip(或*.rar)格式，如 Result_xxx.zip(或 Result_xxx.rar)。
5. 待完成壓縮，即可進行 FTP 上傳 (CIC 已於競賽當日的前 4 天，將 FTP 的 username 及 password 藉電子郵件，寄送到參賽者信箱)。倘若上傳設計檔過程中，有任何問題，請與 CIC 接洽。

請注意!!上傳之 FTP 需切換為二進制模式(binary mode)，且傳輸埠均設為 21 (port:21)。

ftp 的帳號和密碼在賽前已用 email 寄給各參賽者。若有任何問題，請聯絡 CIC

FTP site1 (台灣大學)：iccftp.ee.ntu.edu.tw (140.112.20.92)

FTP site2 (新竹晶片中心)：iccftp.cic.org.tw (140.126.24.18)

FTP site3 (南區晶片中心)：iccftp2.cic.org.tw(140.110.117.9)

6. 若你需要繳交更新版本，請重覆以上步驟，並記得修改 report 檔及壓縮檔的版本編號，因為你無法修改或刪除或覆蓋之前上傳的資料。再次提醒各參賽隊伍，Result_xxx 目錄名稱與 report_xxx.txt 報告檔之“xxx”編號需一致，且依各隊伍上傳設計檔之次數進行編號。評審將以最後上傳的設計檔及報告檔編號進行評分作業!!
7. 建議各參賽隊伍，無論是否完成設計，均需上傳設計檔案!

¹假如參賽者無任何壓縮軟體，您可以至以下連結，下載壓縮軟體試用版。

A. winzip: <http://www.winzip.com>

B. 7zip: <http://www.7-zip.org/download.html>