

A 組-研究所類標準元件數位電路設計

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 **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_a@yuntech.edu.tw

- 信件內容格式如下:
 - 參賽組別：A 組-研究所類標準元件數位電路設計**
 - 參賽編號：(例：10001)**
 - 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

研究所類標準元件數位電路設計

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)兩端的電路設計(S1 及 S2)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)，或是反向將記憶體 RB2 的資料下載到記憶體 RB1。

本傳輸接收電路中，S1 有四只信號輸入(clk、rst、updown、RB1_Q)、四只信號輸出(S1_done、RB1_RW、RB1_A、RB1_D)及二只雙向輸出入信號(sen、sd)。S2 有四只信號輸入(clk、rst、updown、RB2_Q)、四只訊號輸出(S2_done、RB2_RW、RB2_A、RB2_D)、二只雙向輸出入信號(sen、sd)。相關的信號說明，請參考表一和表二。

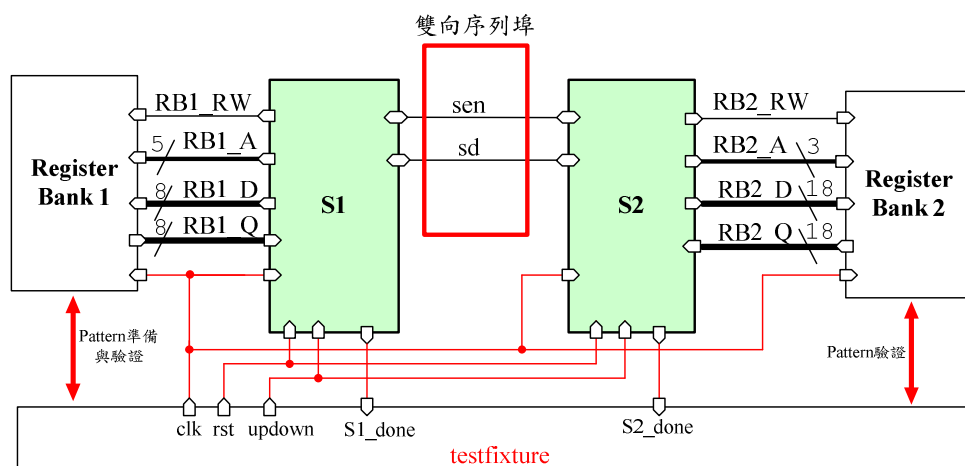
本試題有使用到記憶體模型(memory simulation model)，詳細的記憶體規格描述在 2.2.1，並且其中內含時序檢查，若要避免 RTL 模擬時所產生的時序檢查錯誤(setup or hold violation)，可參考附錄 B 的第 4 點來進行模擬。

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 所提供的輸入指令及正確結果檔來檢查設計是否有達到要求，詳情請參考附錄 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，附上評分所需要的檔案。

2. 設計規格

2.1 系統方塊圖



圖一、系統方塊圖

2.2 輸出入訊號和記憶體描述

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>updown</i>	input	1	系統提供的讀寫控制訊號。 說明：當 updown 為 low 時，RB1 寫入 RB2；為 high 時，RB2 寫入 RB1。
<i>S1_done</i>	output	1	當 S1 寫入 RB1 完成的時後，將 S1_done 設為 1 表示完成。
<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；low 則是進行對 Register Bank1 寫入。
<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>	inout	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>	inout	1	序列傳輸中的一位元雙向資料輸出入埠。

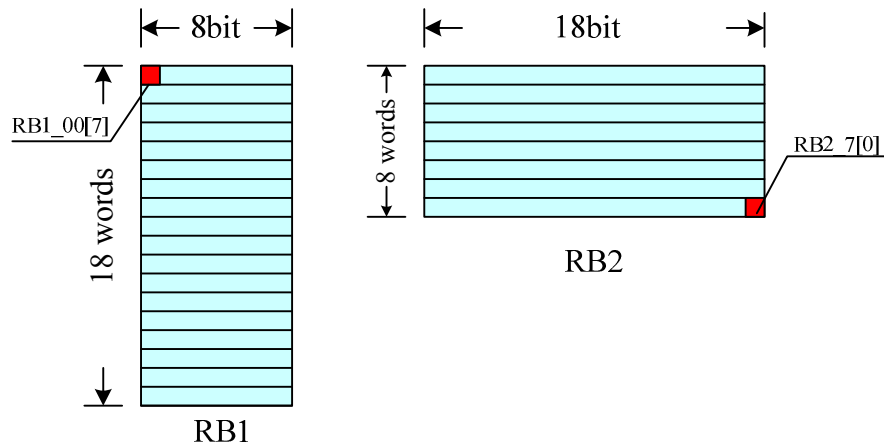
表二、S2 輸入/輸出信號

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>updown</i>	input	1	系統提供的讀寫控制訊號。 說明：當 updown 為 low 時，RB1 寫入 RB2；為 high 時，RB2 寫入 RB1。
<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；low 則是進行對 Register Bank2 寫入。
<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>	inout	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>	inout	1	序列傳輸中的一位元雙向資料輸出入埠。

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都是單埠的 Register File 格式。 Register Bank1，也就是 RB1 模組的記憶體寬度是 8 位元，而記憶體深度是 18 個 word。如果要描述 RB1 中第一個 word 中的 MSB 的話，會用 RB1_00[7]表示，最後一個 word 的 LSB 會用 RB1_17[0]表示。

Register Bank2，也就是 RB2 模組的記憶體寬度是 18 位元，而記憶體深度是 8 個 word。如果要描述 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。請參考圖二。



圖二 RB1 和 RB2 的示意圖

2.3 系統功能描述

本序列傳輸接收電路具有上傳和下載的功能：

當 updown 設定為 0 時，表示進行資料上傳，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。在 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。

當 updown 設定為 1 時，表示進行資料下載，S2 將 RB2 中的資料，轉換成為帶有記憶體位址和資料的封包，傳送到 S1。S1 收到訊號之後，根據封包內的記憶體位址資料存放到 RB1 中。

整個系統操作如下：

- I. 首先 testfixture 會將 RB1 中填滿資料，並且把 updown 設為 0
- II. 當 S1 偵測到 updown 訊號為 0 的時候，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 接收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 送出序列資料時是在 clk **負緣觸發**的時候送出，而 S2 會在 clk **正緣**的時候接收資料(詳細 clk，sen 和 sd 的相關時序關係描述在 2.4)
- III. 當 RB2 所有資料都儲存完成之後，S2 將 S2_done 設為 1 表示已完成上傳的動作。
- IV. testfixture 起動之後，會隨時偵測 S2_done 是否為 1，當確定 S2_done 為 1 時，testfixture 會更新 RB2 內部的資料，並將 updown 訊號會變為 1，要求資料下載。
- V. 這時候 S2 偵測到 updown 訊號為 1，開始將 RB2 的資料轉成序列封包格式回傳到 S1，S1

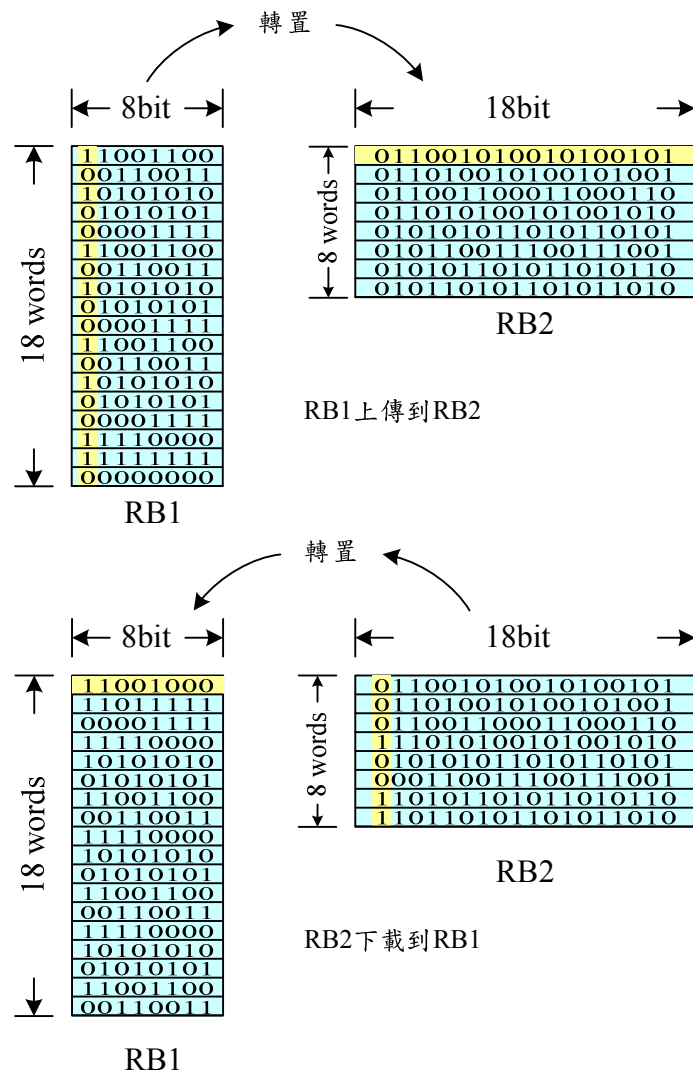
接收封包並根據封包內所存的位址，將資料儲存在 RB1 對應的位址中。

下載時，S2 送出序列資料是在 clk 正緣觸發的時送出，而 S1 將會在負緣觸發的時接收資料

VI. RB1 所有資料儲存完成之後，S1 會將 S1_done 設定為 1 表示完成。

RB1 和 RB2 資料對應關係：

這兩個記憶體模組的 word 都是用來儲存資料的。也就是說原本資料是用 8x18 的方式儲存之後轉置成為 18x8 的方式儲存。可以參考圖三的資料轉置的示意圖。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。



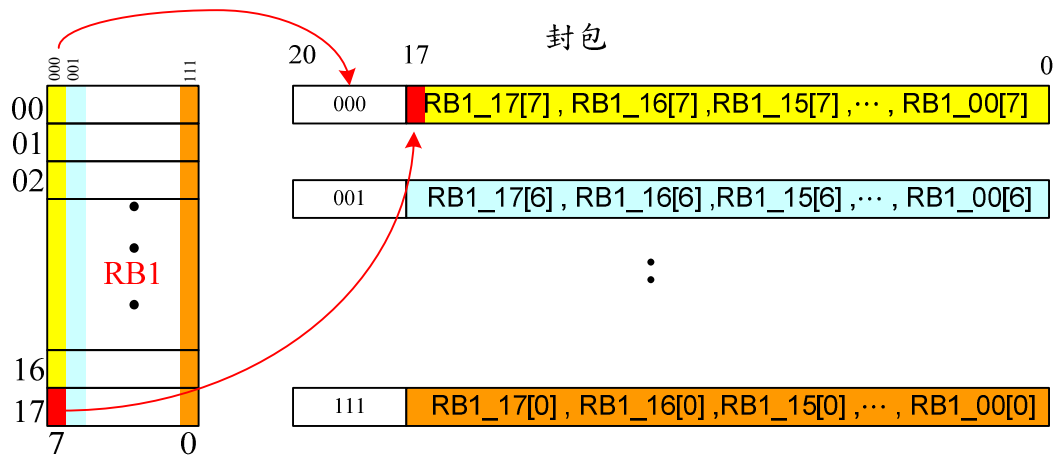
圖三 RB1 和 RB2 轉置範例

2.3.1 封包格式

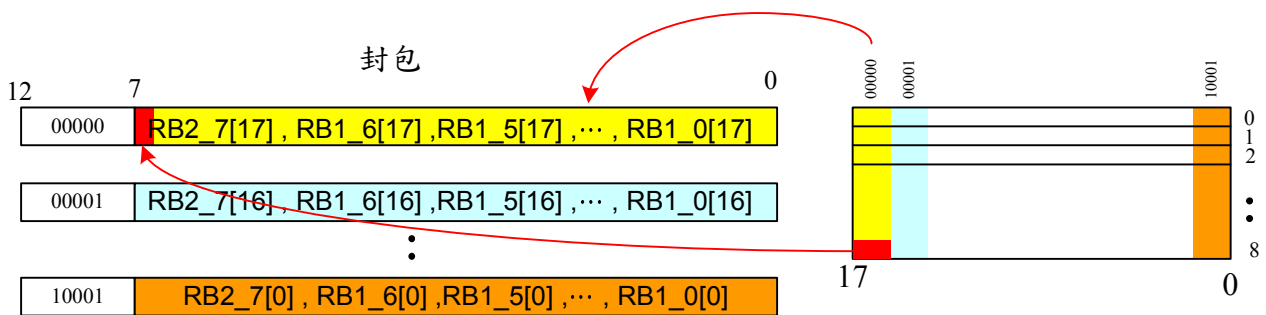
封包格式，最一開始傳輸的是對方記憶體位址，再之後是傳輸資料。不管是記憶體位址或是資料傳輸順序都是 MSB 開始 LSB 結尾，如圖四及圖五所示。

圖四表示上傳時 RB1 以及封包內容的關係。

圖五表示下載時 RB2 以及封包內容的關係，上傳和下載時的封包之長度並不相同。



圖四 RB1 資料與封包的關係圖

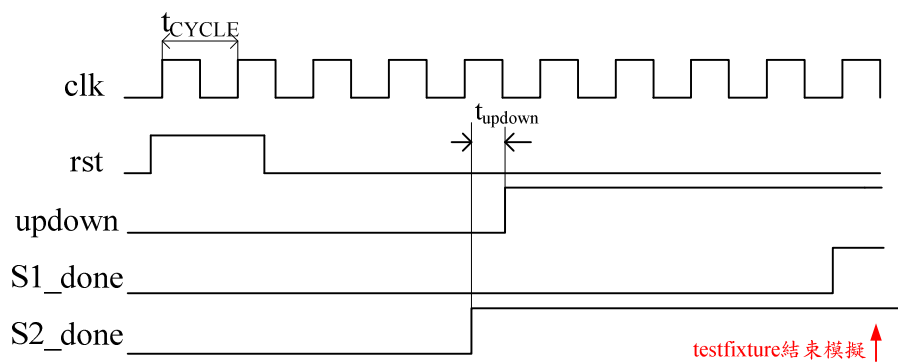


圖五 RB2 資料與封包的關係圖

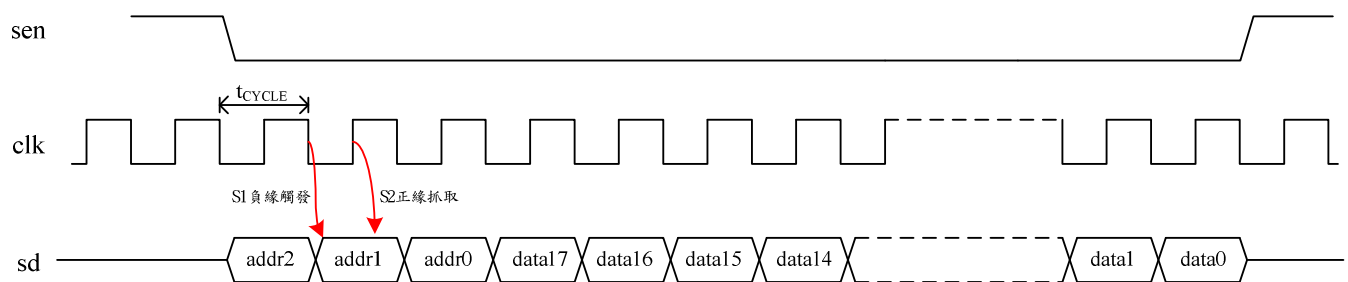
2.3.2 時序規格圖

圖六是系統時序圖，本電路工作時脈為 t_{CYCLE} 是固定在 100ns， t_{updown} 是 updown 的 input setup time，定義為半個時脈週期，也就是 $t_{\text{CYCLE}}/2$ 。rst 會舉為 high 至少維持一個週期以上。testfixture 會檢查 S2_done 是否為 1，當 S2_done 為 1 的時候，經過 t_{updown} 時間之後 updown 會舉為 1。

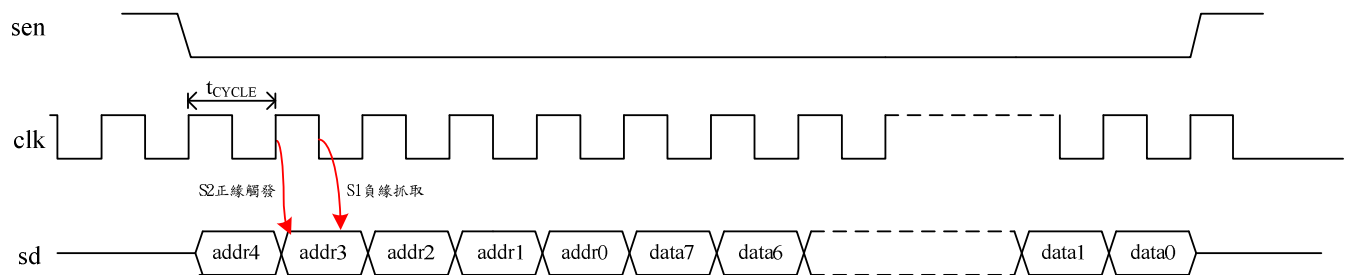
再來 testfixture 會檢查 S1_done 是否為 1，當 S1_done 為 1 時，testfixture 會檢查 RB1 和 RB2 的資料然後結束模擬。圖七是 S1 傳送序列資料到 S2 的時候，序列傳輸埠的波形圖，可以看到 S1 在負緣觸發的時候送出資料，而整個封包有效的情況是在 sen 維持 low 的時候，之後在由 S2 在正緣的時候抓取資料。圖八則是由 S2 傳送序列資料到 S1 時候，序列傳輸埠的波形圖。S2 是正緣觸發送出資料之後在由 S1 在負緣讀取資料。封包有效的情況還是一樣是在 sen 維持 low 的時候。而在每一個封包傳輸的時候，每一個封包需要間隔一個週期以上，如圖九所示。



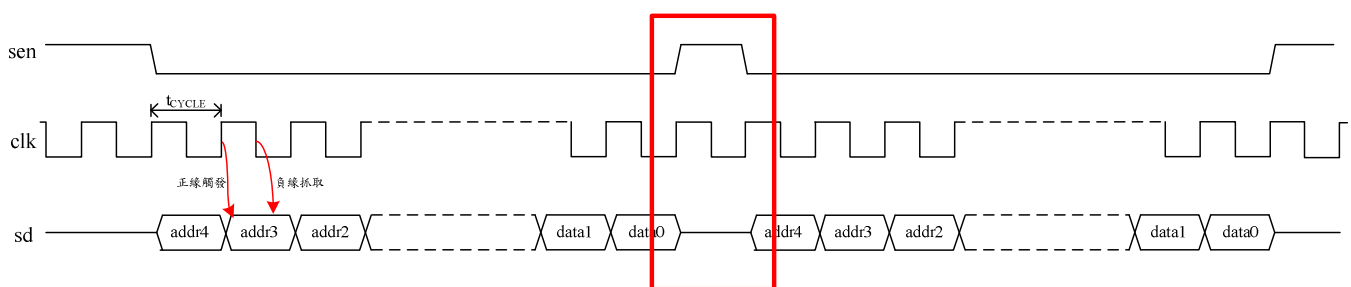
圖六系統時序圖



圖七 S1 傳輸 S2 操作關係圖



圖八 S2 傳輸 S1 操作關係圖



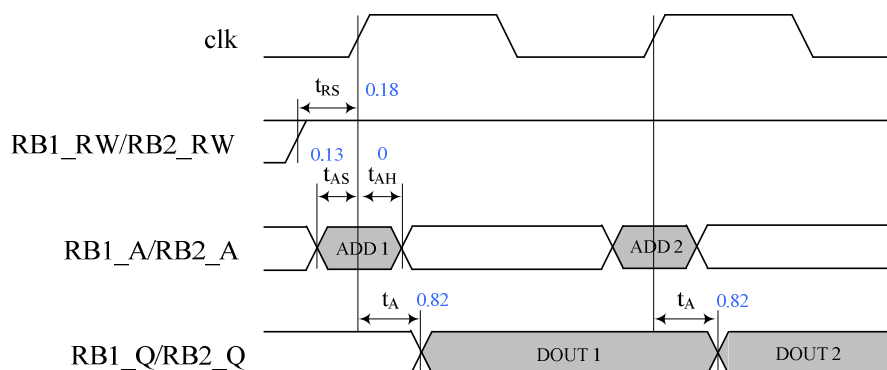
圖九 連續封包的傳輸關係圖 - S2 傳輸到 S1 為例

由於 sen 和 sd 資料信號為雙向埠，因此在 verilog coding 時可能會需要使用到下列的語法：

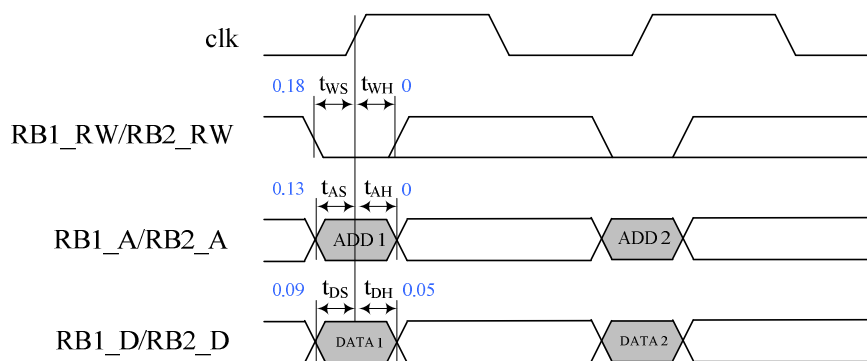
```
assign output = (enable) ? input : 'bz;
```

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，其資料讀取與寫入之波形時序圖，如下面圖十、圖十一所示。**特別注意若沒有要進行寫入時，請將 RB1_RW/RB2_RW 保持為 high。**最後，時序的數值整理於表三。



圖十、RB1/RB2 資料讀取波形



圖十一、RB1/RB2 資料寫入波形

表三、RB1/RB2 記憶體時序參數表

Symbol	Description	Min.	Unit
t_A	Access Time	0.82	ns
t_{RS}	Read Enable Setup Time	0.18	ns
t_{WS}	Write Enable Setup Time	0.18	ns
t_{WH}	Write Enable Hold Time	0	ns
t_{AS}	Address Setup Time	0.13	ns
t_{AH}	Address Hold Time	0	ns
t_{DS}	Data Setup Time	0.09	ns
t_{DH}	Data Hold Time	0.05	ns

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，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 CIC 所提供的標準設計結果為準。

CIC 將本試題區分為下面三個等級來作為功能完成度之評分，完成度越高者優先錄取。若為同一等級則以檔案上傳時間來評分，且對每個等級分別進行 RTL 與 gate-level simulation：

1. A 等級：完成測試樣本一至二 (tb1 & tb2)之 RTL 與 gate-level simulation
2. B 等級：完成測試樣本一至二 (tb1 & tb2)之 RTL simulation
3. C 等級：完成測試樣本一 (tb1)的上傳功能 之 RTL simulation

B 等級至 C 等級雖不須進行 synthesis，但 RTL code 須為 synthesizable RTL code。

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

競賽所提供的設計軟體與版本如下表四。驗證評分時，係以所列軟體及版本作為驗證依據。

表四、設計軟體版本

Functionality	Corresponding EDA tools
Logic Simulator	nc-verilog v8.1 modelsim v6.3c vcs v2009.06
Logic Synthesizer	design-compiler v2009.06-sp4

附錄 B 設計檔案說明

1. 下表七為主辦單位所提供各參賽者的設計檔案

表五、設計檔

檔名	說明
testfixture.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信號
S1.v	參賽者所使用的設計檔，已包含系統輸/出入埠之宣告
S2.v	參賽者所使用的設計檔，已包含系統輸/出入埠之宣告
RB1.v	Register Bank 1 模擬檔
RB2.v	Register Bank 1 模擬檔
synopsys_dc.setup	Design Compiler 初始設定範例檔案
S1.sdc	Design Compiler 電路合成規範檔
S2.sdc	Design Compiler 電路合成規範檔
tb1_RB1_in.dat	測試樣本一之 RB1 輸入檔案
tb1_RB1_goal.dat	測試樣本一之 RB1 比對檔
tb1_RB2_goal.dat	測試樣本一之 RB2 比對檔
tb2_RB1_in.dat	測試樣本二之 RB1 輸入檔案
tb2_RB1_goal.dat	測試樣本二之 RB1 比對檔
tb2_RB2_goal.dat	測試樣本二之 RB2 比對檔
report.000	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/入埠宣告如下所示：

```

module S1(clk, rst, updown, S1_done, RB1_RW, RB1_A, RB1_D, RB1_Q, sen, sd);
    input clk, rst, updown;
    output S1_done, RB1_RW;
    output [4:0] RB1_A;
    output [7:0] RB1_D;
    input [7:0] RB1_Q;
    inout sen, sd;
endmodule

```

```

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

```

3. 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：

- ncverilog 指令範例如下：
ncverilog testfixture.v S1.v S2.v +define+tb1
- 若使用 modelsim，則是在 compiler verilog 時，使用下面指令：
vlog testfixture.v S1.v S2.v +define+tb1
- 若使用 vcs，則是在 compiler verilog 時，使用下面指令：
vcs -R +v2k testfixture.v S1.v S2.v +define+tb1
- 上述指令中 ***+define+tb1*** 指的是使用第一組測試樣本模擬，若須使用其它測試樣本請自行修改此參數。以第二組測試樣本為例：***+define+tb2***。
- 關於模擬時使用的一些記憶體，因已經以 include 方式加在 testfixture.v 裏，所以不需加在模擬指令裏。
- 若 RTL 模擬時，要避免時序檢查以減少錯誤訊息，可於模擬指令中加入 ***+notimingchecks***
 範例如：***ncverilog testfixture.v S1.v S2.v +define+tb1 +notimingchecks***
vlog testfixture.v S1.v S2.v +define+tb1 +notimingchecks
vcs -R +v2k testfixture.v S1.v S2.v +define+tb1 +notimingchecks

4. dump 波形檔請參考下列指令：

- ncverilog 指令範例如下(請先 source verdi 的環境設定檔)：
ncverilog testfixture.v S1.v S2.v +define+tb1+FSDB +access+r
- modelsim 使用者，請直接使用內建波形來進行除錯。
- vcs 指令範例如下：
vcs -R +v2k testfixture.v S1.v S2.v +define+tb2+FSDB -P
<Your_Verdi_Path>/Verdi.tab <Your_Verdi_Path>/pli.a

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)gate-level design，即由合成軟體所產生的 gate-level netlist，以及對應的 SDF 檔；(3)report file，參賽隊伍必須依照自己的設計內容，撰寫 report.000 檔，以方便主辦單位進行評分，report.000 的格式如圖十所示。(report 檔以後三碼序號表示版本，若繳交檔案更新版本，則新版的 report 檔檔名為 report.001，依此類推)

表六、評分用檔案

RTL category		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
N/A	report.xxx	design report
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.vg	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

隊號(Team number): 999999

完成模擬之測試樣本等級(level of test pattern) : A

--- RTL category---

使用之 HDL 模擬器名稱(HDL simulator): nc-verilog

RTL 檔案名稱(RTL filename): S1.v, S2.v, 以及使用到的子模組檔案...

--- Pre-layout gate-level ---

gate-level 檔案名稱(gate_level filename): S1_syn.vg, S2_syn.vg

gate-level sdf filename: S1_syn.sdf, S2_syn.sdf

design compiler 合成資料庫(dc library): S1_syn.ddc, S2_syn.ddc

-----注意事項(annotation)-----

(其餘注意事項依各參賽隊伍的需求填寫)

圖十、report.000 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。並且，提交的設計檔案，需要經過壓縮於同一個資料夾下，步驟如下：

1. 建立一個 result_xxx 資料夾。其中“xxx”表示繳交版本。例如“000”表示為第一次上傳；“001”表示為第二度上傳；002 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. 執行 tar 指令將 result_xxx 資料夾包裝起來，tar 的指令範例如下：
tar cvf result_xxx.tar result_xxx
其中 xxx 表示繳交版本
執行完後應該會得到 result_xxx.tar 的檔案
4. 使用 ftp 將 result_xxx.tar 及 report.xxx 一併上傳至 CIC 提供的 ftp server，result_xxx.tar 與 report.xxx 之“xxx”編號需一致，評審將以**最後上傳的設計檔及報告檔編號進行評分作業**。

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)

5. 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 tar 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。