B 組-大學/研究所標準元件數位電路設計

初賽時間為 3 月 19 日(三) 8:30~20:30，初賽當日請密切注意競賽網頁公告及 Facebook “IC 競賽專頁”(https://www.facebook.com/IcContest2014ByNcu)，所有最新訊息將做即時動態更新。

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。

參賽作品在今天 20:30 前務必根據初賽前寄發之 ftp 帳號密碼上傳至國家晶片中心之相關 ftp site。

■ 初賽登錄: 請將您的隊伍參賽資料 e-mail 至 B.icdesign.ncu@gmail.com

信件主旨:登錄(ID: B?????)(請填上自己的報名 ID)

信件內容:

組別: B 組
ID: B????? (例: B00001)
姓名: 李大華、王小明

2014 IC Design Contest Preliminary

研究所類標準元件數位電路設計

Serial Transmitter and Data Arrange Controller

1. 問題描述

請完成一個系統，其訊號界面如圖一所示，內容包含一序列傳輸介面處理電路(Serial Transmitter Interface, STI)及一資料排列控制電路(Data Arrange Controller, DAC) (如圖二)。STI 電路動作為從並列埠進行資料輸入處理後由序列埠將處理完成之資料以序列輸出。DAC 電路之功能為將經 STI 電路處理完成後之序列資料進行排列後分別寫入指定記憶體。

本試題電路中，共有九只信號輸入(clk、rst、load、pi_data、pi_length、pi_fill、pi_msb、pi_low、pi_end)、十三只信號輸出（so_data, so_valid, oem_finish、oem_addr、oem_dataout、odd1_wr、even1_wr、odd2_wr、even2_wr、odd3_wr、even3_wr、odd4_wr、even4_wr）。參賽者須將 STI 及 DAC 處理電路設計於同一設計檔(STI_DAC)內。相關信號說明，請參考表一。

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。

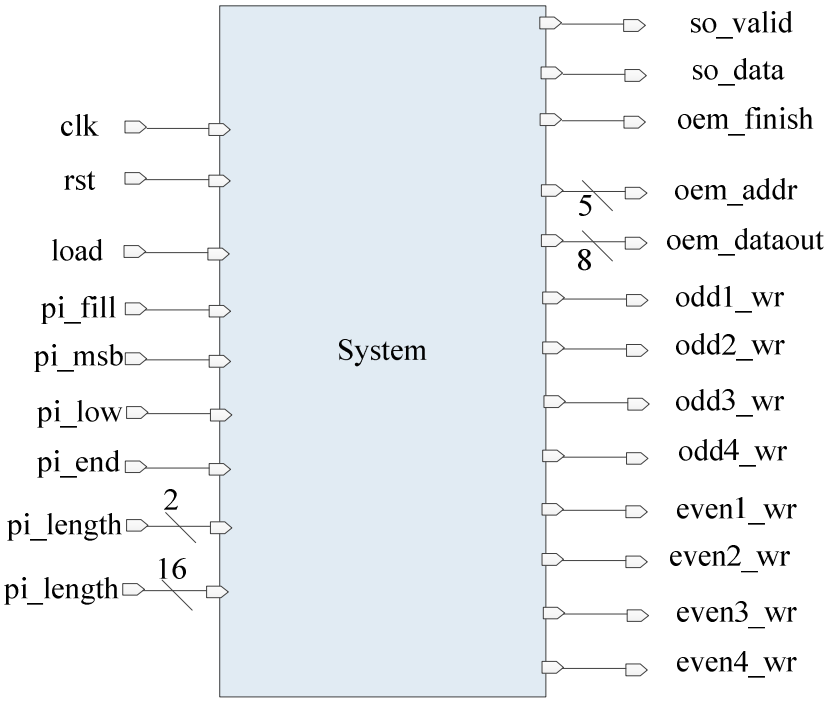
本次 IC 設計競賽比賽時間為上午 08:30 到下午 08:30。當 IC 設計競賽結束後，CIC 會根據第三節中的評分標準進行評分。為了評分作業的方便，各參賽隊伍應參考附錄 D 中所列的要求，附上評分所需要的檔案。

2. 設計規格

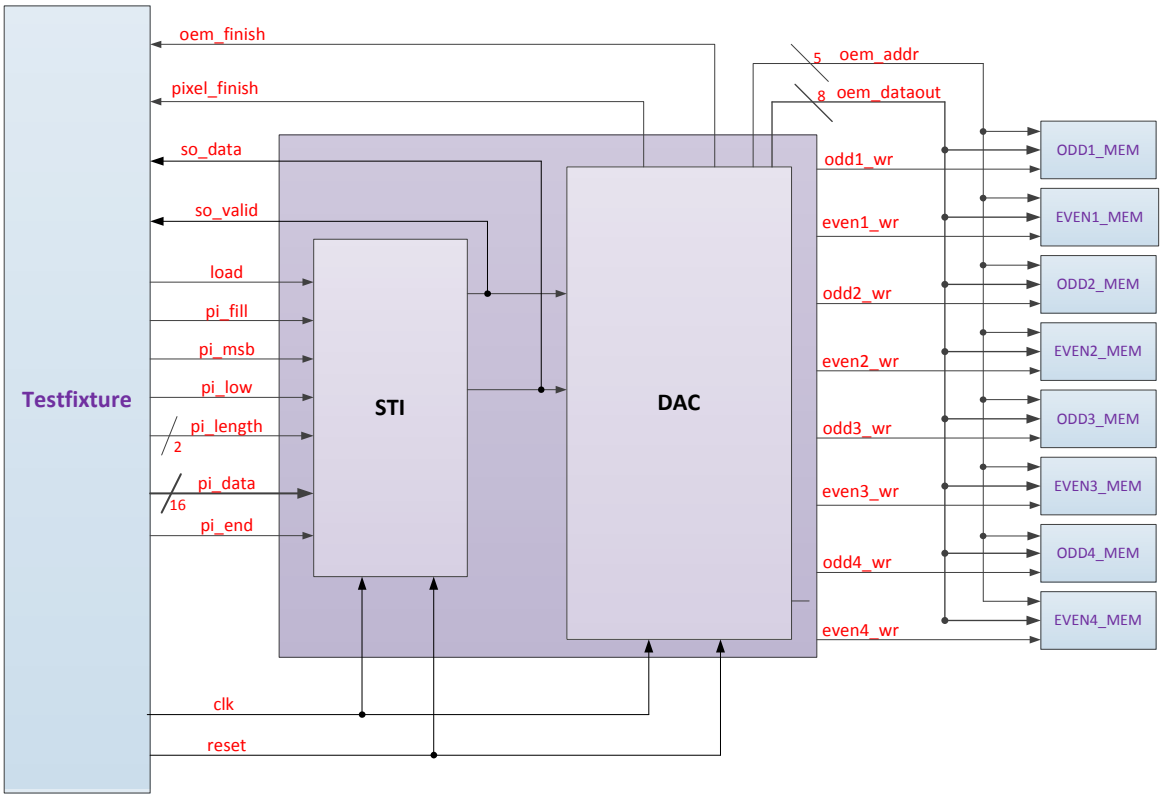
請注意：

1. 此次**top module名稱及檔案名稱、大小寫須完全符合附錄B規範**，若有引入其它模組、檔案請自行寫在設計檔內，測試檔不允許任何修改否則不予計分。
2. 最後評分方式為使用最後上傳檔案版本評分，並以最後上傳檔案版本時間為依據，請參考3.評分標準。

2.1 系統方塊圖



圖一、系統方塊圖



圖二、系統架構參考圖

2.2 輸出入訊號和記憶體描述

表一、STI_DAC 輸入/輸出信號

信號名稱	輸入/輸出	位元寬度	說明
clk	input	1	系統提供的時脈信號。
reset	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。
load	input	1	系統提供的讀取控制訊號。 說明：訊號寬度持續一個時脈週期。當 load = 1 時且經時脈訊號正緣觸發時，表示並列資料輸入埠及序列控制訊號為有效。
pi_data	input	16	十六位元並列資料輸入埠。
pi_length	input	2	序列資料輸出長度設定信號。 說明：當此訊號呈現 2'b00 時，表示序列輸出為 8bits 資料輸出。 當此訊號呈現 2'b01 時，表示序列輸出為 16bits 資料輸出。 當此訊號呈現 2'b10 時，表示序列輸出為 24bits 資料輸出。 當此訊號呈現 2'b11 時，表示序列輸出為 32bits 資料輸出。
pi_fill	input	1	序列資料輸出模式設定信號。
pi_msb	input	1	序列輸出順序控制訊號。
pi_low	input	1	序列低位元輸出致能訊號。
so_data	output	1	序列資料輸出埠。
so_valid	output	1	序列資料輸出致能訊號。 說明：當此信號為 1 時，表示 so_data 傳輸的資料被認為是有效的。
pi_end	input	1	並列資料輸入結束旗標。 說明：當此訊號為 1 時，表示測試樣本檔將不再向 STI_DAC 輸入任何資料；當此訊號為 0 時，表示測試檔樣本仍可能會對 STI_DAC 進行資料輸入。
oem_finish	output	1	OM 及 EM 記憶體共用寫入完成指示訊號。 說明：當記憶體 ODD1_MEM~ODD4_MEM 及 EVEN1_MEM~EVEN4_MEM 完成資料寫入時，將 oem_finish 設定為 1，則測試樣本檔將開始進行驗證記憶體內容；預設值應設定為 0。
oem_addr	output	5	OM 及 EM 記憶體共用五位元位址埠。
oem_dataout	output	8	OM 及 EM 記憶體共用八位元資料埠。
odd1_wr	output	1	ODD1_MEM 記憶體資料寫入致能訊號。
even1_wr	output	1	EVEN1_MEM 記憶體資料寫入致能訊號。
odd2_wr	output	1	ODD2_MEM 記憶體資料寫入致能訊號。
even2_wr	output	1	EVEN2_MEM 記憶體資料寫入致能訊號。
odd3_wr	output	1	ODD3_MEM 記憶體資料寫入致能訊號。
even3_wr	output	1	EVEN3_MEM 記憶體資料寫入致能訊號。
odd4_wr	output	1	ODD4_MEM 記憶體資料寫入致能訊號。
even4_wr	output	1	EVEN4_MEM 記憶體資料寫入致能訊號。

2.3 系統功能描述

本序列傳輸介面處理電路(STI)功能如下：

當 reset 結束後。每當 load = 1 且經時脈訊號正緣觸發時，表示 STI 輸入訊號為有效，STI 將依據控制訊號(pi_length、pi_fill、pi_msb、pi_low)之設定將 pi_data 輸入訊號進行相對應之並列轉序列資料處理，處理完成後將 so_valid 拉成 1 表示有效資料輸出，並將處理完成之資料由 so_data 依序送出。當 load = 0 時，表示輸入資料無效，STI 將不進行任何動作。

以下文中 MSB 代表 Most Significant Bit；LSB 代表 Least Significant Bit。

並列轉序列資料處理規範如下:

I. 當 pi_msb 輸入為 1 時，表示序列輸出由序列輸出緩衝資料的 MSB 開始；當 pi_msb 輸入為 0 時，表示序列輸出由序列輸出緩衝資料的 LSB 開始。(詳細處理關係描述在 2.3.1)

II. pi_fill 控制訊號僅在 pi_length 設定為 24bits 及 32bits 序列輸出時有效。當 pi_fill 輸入為 1 時，表示 pi_data 並列資料對齊於序列輸出緩衝資料的 MSB，而序列輸出緩衝資料的其餘位元都為 0；當 pi_fill 輸入為 0 時，表示 pi_data 並列資料對齊於序列輸出緩衝資料的 LSB，而序列輸出緩衝資料的其餘位元都為 0。(詳細處理關係描述在 2.3.2)

III. pi_low 僅在 pi_length 設定為 8bits 序列資料輸出時有效。當 pi_low 輸入為 1 時，表示序列輸出緩衝資料為 pi_data 並列資料的高位元組共 8bits；當 pi_low 輸入為 0 時，表示序列輸出緩衝資料為 pi_data 並列資料的低位元組共 8bits。(詳細處理關係描述在 2.3.3)

IV. pi_length 設定序列輸出訊號長度(詳細處理關係描述在 2.3.4)。

V. pi_length 搭配 pi_fill 或 pi_low 可決定序列輸出緩衝資料的處理格式，處理後的序列輸出緩衝資料搭配 pi_msb 可決定由序列輸出緩衝資料的 MSB 或 LSB 開始序列輸出。

當序列輸出緩衝資料處理完畢後，將 so_valid 輸出為 1 並將序列資料由 so_data 依序送出。

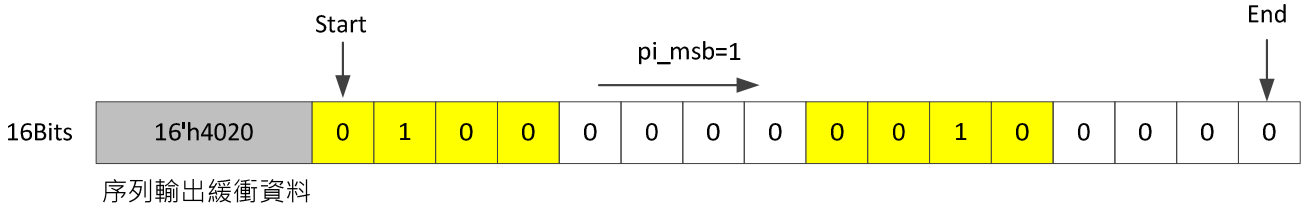
每筆 pi_data 會輸入 16bits 的並列資料，而每筆 so_data 輸出為 8 或 16 或 24 或 32 筆序列輸出，每筆 pi_data 處理完成後，須將 so_valid 輸出為 0，之後 testfixture 將會自動把下一筆待處理資料輸入。

本資料排列控制電路（DAC）電路功能如下：

將 STI 序列傳輸介面處理電路輸出的有效序列訊號(so_valid=1 時的 so_data 輸出)，依序以每 8 位元資料構成 1 筆資料存入 1 個資料記憶體位址。因此第 1 筆 8 位元資料之記憶體位址為 0、第 2 筆 8 位元資料之記憶體位址為 1…以此類推，依序將如此每八個位元的有效序列輸出內容儲存至資料記憶體中，資料記憶體構成方法可參考 2.3.8 及圖十五.說明。本題目限定資料記憶體大小及其位址最多不超過 256 筆，若有效序列輸出構成之資料筆數少於 256 筆，則其餘資料記憶體內容寫入 8'h00。DAC 控制電路須將上述資料記憶體內容，分別取出特定位址之資料，並分別存入 OM 記憶體（ODD1_MEM、ODD2_MEM、ODD3_MEM、ODD4_MEM ）共四個(詳細處理關係描述在 2.3.5)、及 EM 記憶體（EVEN1_MEM、EVEN 2_MEM、EVEN 3_MEM、EVEN 4_MEM ）共四個(詳細處理關係描述在 2.3.6)。這八個記憶體都儲存完成後將 oem_finish 輸出為 1，完成 DAC 電路動作。*關於上述 DAC 處理過程中所使用的資料記憶體，參賽者可自行決定是否使用。*

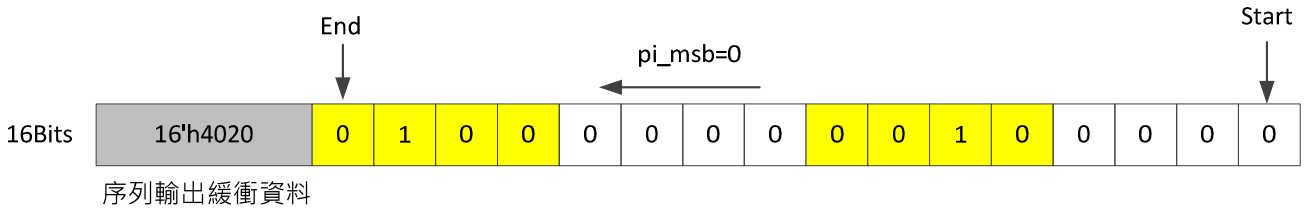
2.3.1 最高位元優先功能描述（pi_msb）

當 pi_msb 輸入為 1 時，表示 so_data 序列輸出由序列輸出緩衝資料的 MSB 開始，如圖三.（範例使用 16 bits 說明）所示，若 pi_data 輸入為 16'h4020，當 pi_msb 輸入為 1 時，其 16 bits 序列輸出緩衝資料由 so_data 依序輸出 0,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0。



圖三、最高位元優先傳輸資料格式（pi_msb=1）

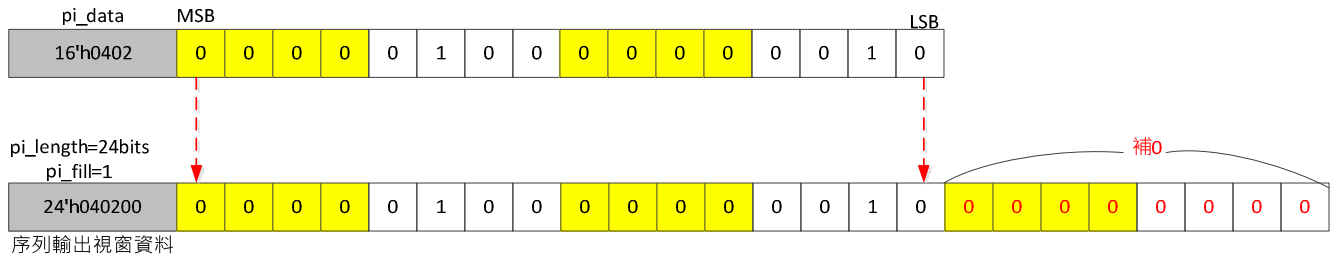
當 pi_msb 輸入為 0 時，表示 so_data 序列輸出由序列輸出緩衝資料的 LSB 開始，如圖四.（範例使用 16 bits 說明）所示，若 pi_data 輸入為 16'h4020，當 pi_msb 輸入為 0 時，其 16 bits 序列輸出緩衝資料由 so_data 依序輸出 0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0。



圖四、最高位元優先傳輸資料格式（pi_msb=0）

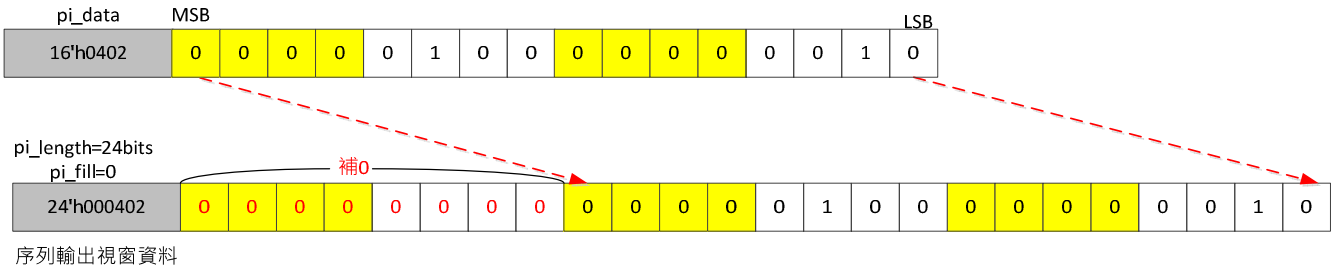
2.3.2 位元填滿模式功能描述（pi_fill）

pi_length 設定為 24bits 及 32bits 序列輸出時，且當 pi_fill 輸入為 1 時，表示 pi_data 並列資料與序列輸出緩衝資料為由 MSB 對齊，其餘位元都補 0，如圖五.範例所示，pi_data 輸入 16'h0402 的資料時，pi_length=2'b10（24bits 資料輸出）、pi_fill=1 時，則序列輸出緩衝資料的資料 so_data 依序為 24'h040200。



圖五、位元填滿模式資料格式（pi_fill=1）

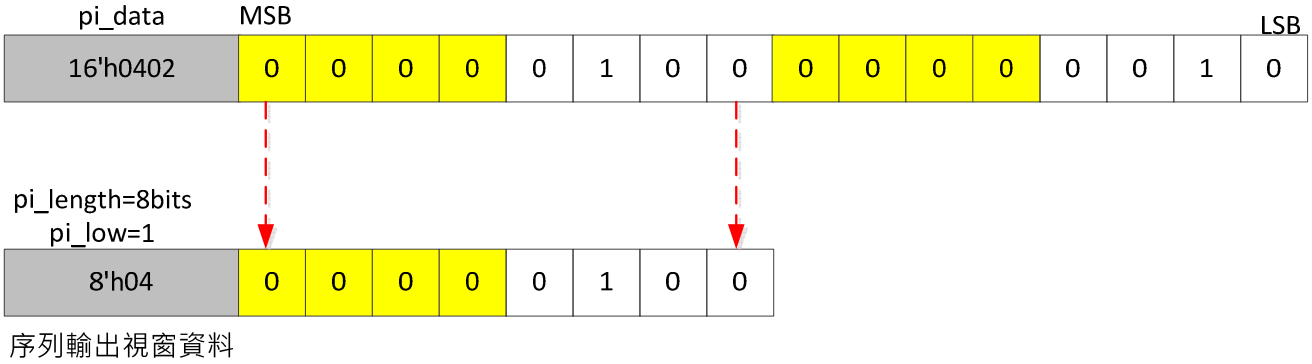
當 pi_fill 輸入為 0 時，表示 pi_data 並列資料與序列輸出緩衝資料為由 LSB 對齊，其餘位元都補 0。如圖六.範例所示，pi_data 輸入 16'h0402 的資料時，pi_length=2'b10（24bits 資料輸出）、pi_fill=0 時，則序列輸出緩衝資料的資料 so_data 依序為 24'h000402。



圖六、位元填滿模式資料格式（pi_fill=0）

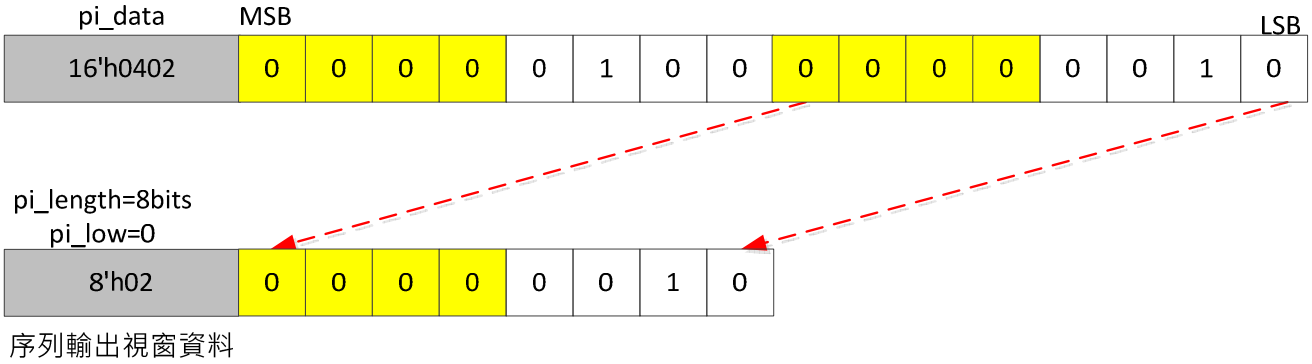
2.3.3 低位元組資料致能功能描述（pi_low）

pi_low 僅在 pi_length 設定為 8bits 序列資料輸出時有效。當 pi_low 輸入為 1 時，表示序列輸出緩衝資料為 pi_data 並列資料的高位元組共 8bits。如圖七.範例所示，pi_data=16'h0402，pi_length=2'b00（8bits），pi_low = 1，則輸出 pi_data 的高位元組 8bits（8'h04）



圖七、低位元組資料模式資料格式（pi_low =1）

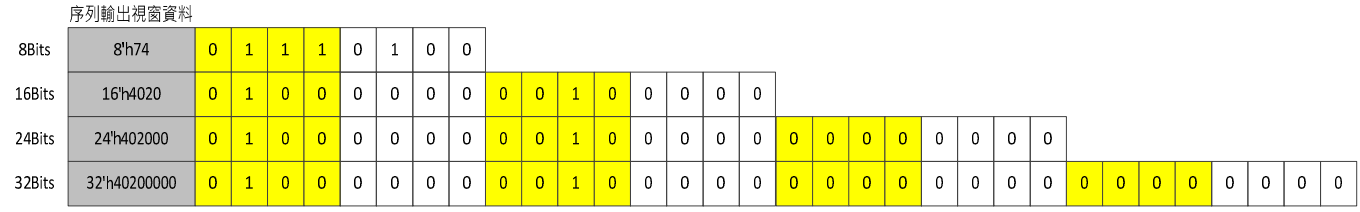
當 pi_low 輸入為 0 時，表示序列輸出緩衝資料為 pi_data 並列資料的低位元組共 8bits。如圖八.範例所示，pi_data=16'h0402，pi_length=2'b00（8bits），pi_low = 0，則輸出 pi_data 的低位元組 8bits（8'h02）。



圖八、低位元組資料模式資料格式（pi_low =0）

2.3.4 序列輸出訊號長度功能描述（pi_length）

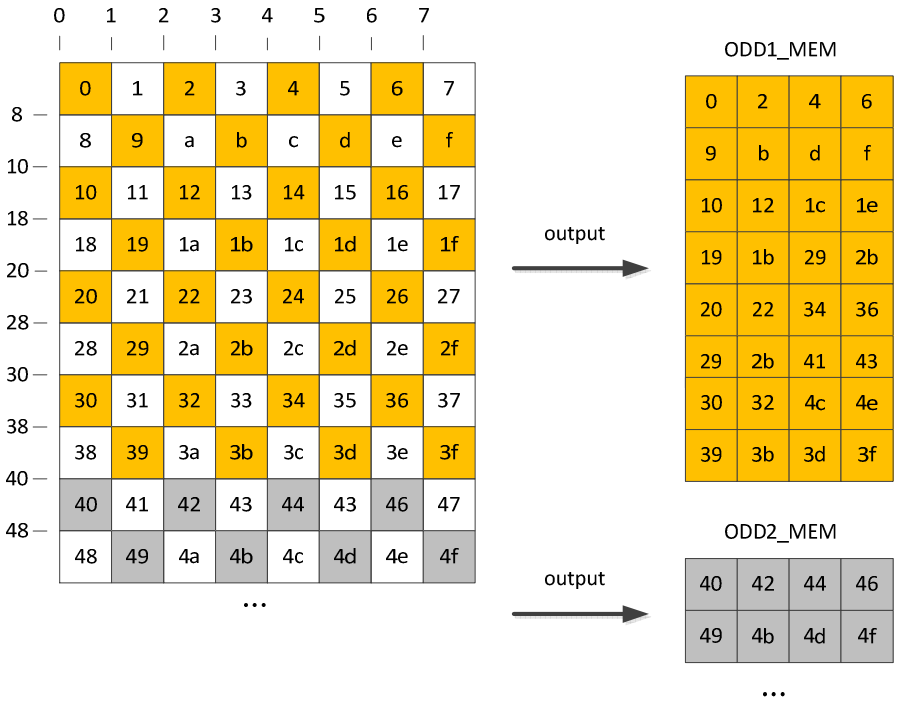
序列輸出資料格式共有 8 位元、16 位元、24 位元及 32 位元資料格式，如圖九所示，依 2.3.1~2.3.3 指令需求，利用 so_data 將序列資料輸出。



圖九、序列傳輸資料長度

2.3.5 OM 記憶體

OM 記憶體包含 ODD1_MEM、ODD2_MEM、ODD3_MEM、ODD4_MEM 等四個記憶體。
DAC 控制電路將資料記憶體特定位址之資料取出，取出規則為第一列取 0,2,4,6 位址，第二列取依序 9,b,d,f 位址 ，第三列則取 10,12,14,16 位址……其餘依此類推。因此依序將 0,2,4,6,9,b,d,f,10….fd,ff 位址資料寫入至 OM 記憶體，每個 OM 記憶體為 8 bits x32，若超過 32 個位址空間，須自動分頁儲存至下一個 OM 記憶體，例如：ODD1_MEM 儲存滿 32 筆 8 位元資料後，再依序儲存至 ODD2_MEM、ODD3_MEM、ODD4_MEM。如圖十.所示。

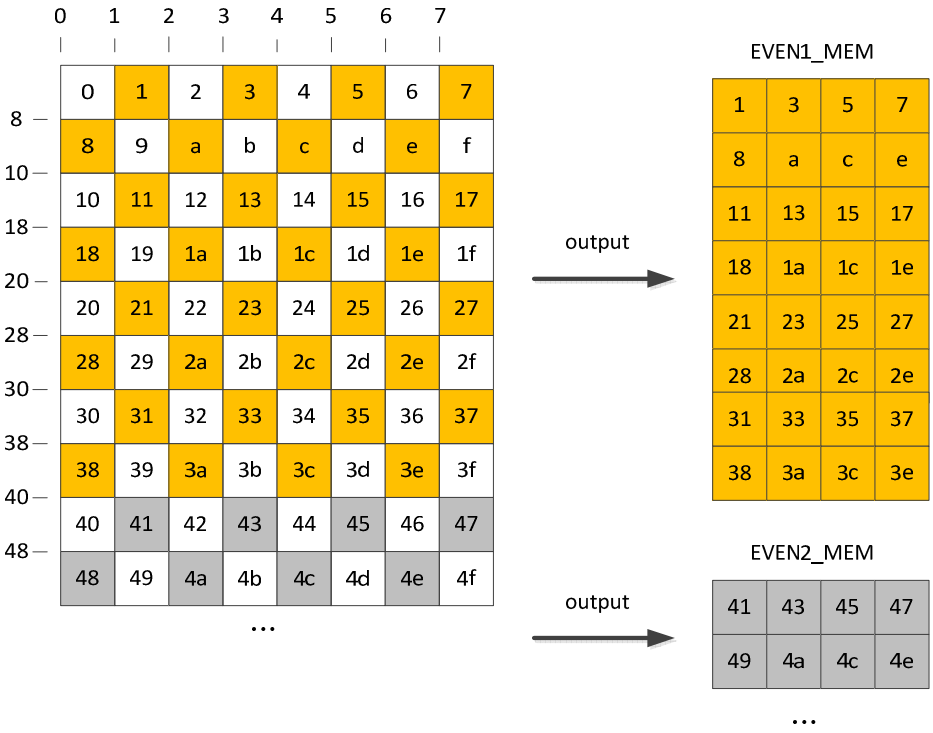


圖十、OM 記憶體資料寫入格式

2.3.6 EM 記憶體

EM 記憶體包含 EVEN1_MEM、EVEN 2_MEM、EVEN 3_MEM、EVEN 4_MEM 等四個記憶體。
DAC 控制電路將資料記憶體特定位址之資料取出，取出規則為第一列取 1,3,5,7 位址，第二列取依序 8,a,c,e 位址 ，第三列則取 11,13,15,17 位址……其餘依此類推。因此依序將 1,3,5,7,8,a,c,e,11….fc,fe 位址資料寫入至 EM 記憶體，EM 記憶體為 8 bits x32，若超過 32 個位址空間，須自動分頁儲存至下一個 EM 記憶體，例如：EVEN1_MEM 儲存滿 32 筆 8 位元資料後，再依序儲存至 EVEN2_MEM、

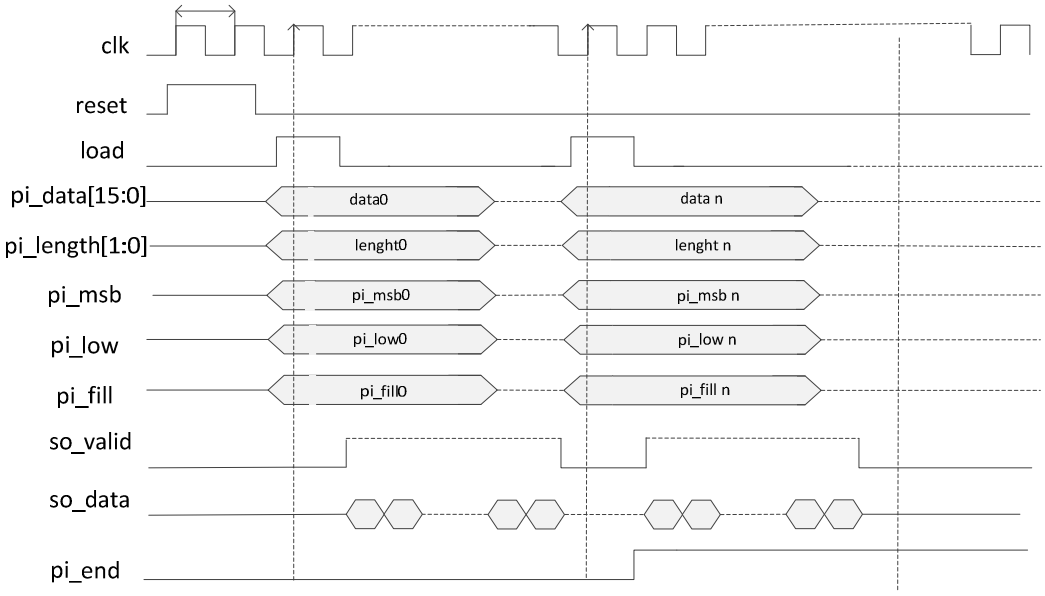
EVEN3_MEM、EVEN4_MEM。如圖十一.所示。



圖十一、EM 記憶體資料寫入格式

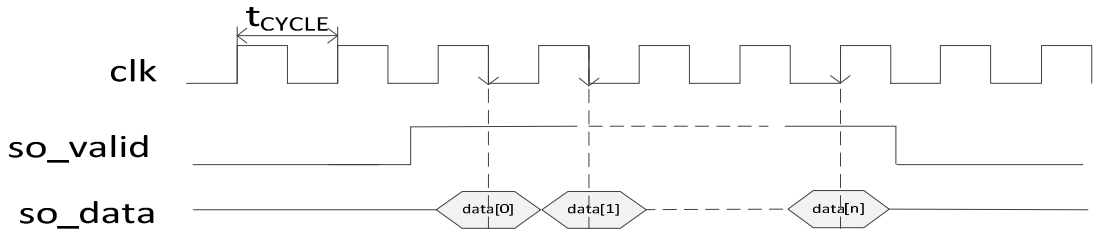
2.3.7 時序規格圖

圖十二.是系統初始時序圖，其中 reset 將維持至少一個 t_{CYCLE} 的 1，待 reset 輸入為 0 之後，testfixture 將開始輸入並列資料，此時若 load = 1 且經 clk 的 rising edge 觸發則表示當下的控制指令為有效指令，STI 電路須針對此有效指令進行並列轉序列處理，load 訊號將只會維持一個 t_{CYCLE} 為 1。當 testfixture 偵測到 so_valid 由 1 轉 0 時，將會隨後送出下一筆並列輸入訊號；待最後一筆 pi_data 輸入後，testfixture 隨即會將 pi_end 輸出為 1 表示不會再有資料輸入。



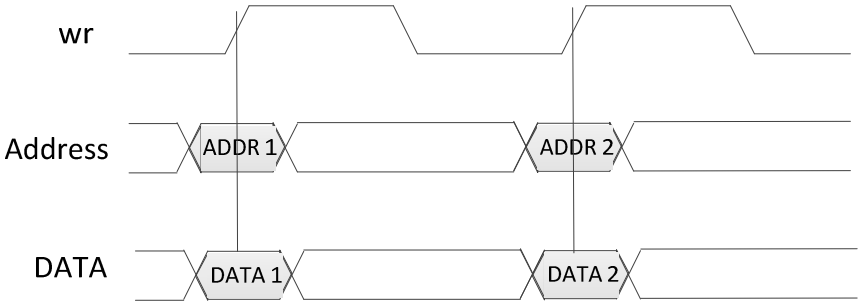
圖十二、主控端訊號傳輸時序圖

STI 進行並列轉序列輸出時序可參考下列圖十三.所示。每筆 pi_data 資料處理完成後利用 so_data 序列輸出，並且把 so_valid 輸出為 1，表示目前輸出為有效的。而 testfixture 將會在偵測到 so_valid = 1 且 clk 的 falling edge 時進行序列資料輸出比對。每一筆並列資料輸入處理後的序列輸出必須為連續完整輸出。



圖十三、序列訊號資料輸出時序圖

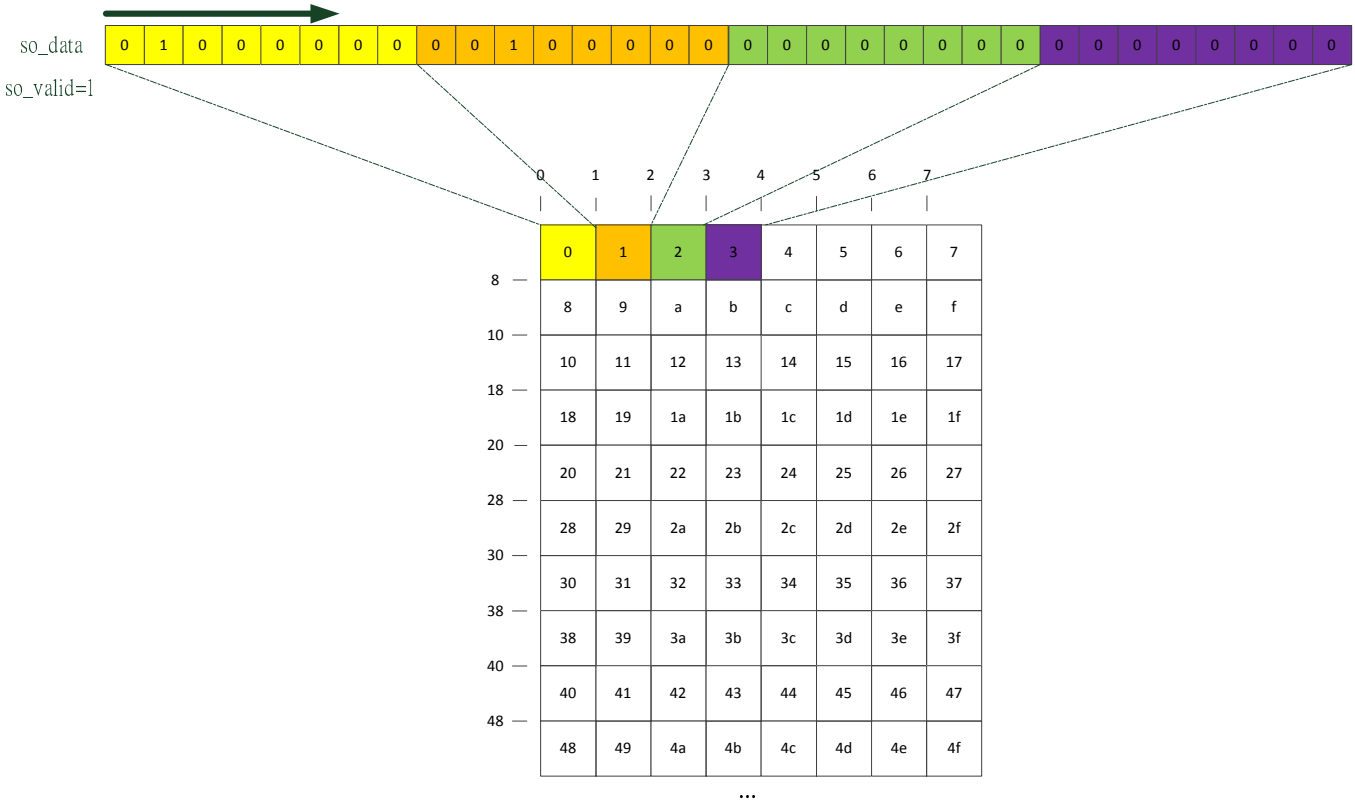
DAC 電路須將所有的有效序列輸出資料依 2.3.5 及 2.3.6 規範寫到指定記憶體內。寫入方式如圖十四.所示，當內建於測試檔的 ODD1_MEM~ODD4_MEM 及 EVEN1_MEM~EVEN4_MEM 共八顆 OM 及 EM 記憶體分別偵測到其各別的寫入致能訊號(odd1_wr~odd4_wr 及 even1_wr~even4_wr 共八條)的 rising edge 時，就會將當下 oem_dataout 資料埠內容寫入 oem_addr 位址埠所指定之記憶體位址內。因此資料埠及位址埠為這八顆記憶體共用，參賽者須謹慎處理八條寫入致能訊號之動作。當所有 OM 及 EM 記憶體位址都寫入完成後，請將 oem_finish 訊號輸出為 1，testfixture 將在偵測到 oem_finish 訊號為 1 後立刻進行記憶體內容驗證。



圖十四、記憶體資料寫入時序

2.3.8 資料記憶體構成方法

以下圖十五.是資料記憶體構成方法說明圖。由有效序列輸出每 8bits 構成一個位址內容。



圖十五、資料記憶體構成方法

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四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：

1. A 等級：完成 STI_DAC 電路的測試樣本 1 及 2 之 RTL 與 gate-level simulation(操作時脈週期限定須在 100ns 以下)，並且合成後 Cell Area 小於 **20000**。
2. B 等級：完成 STI_DAC 電路的測試樣本 1 及 2 之 RTL 與 gate-level simulation(操作時脈週期限定須在 100ns 以下)。
3. C 等級：完成 STI_DAC 電路的測試樣本 1 及 2 之 simulation。
4. D 等級：完成測試樣本 1 及 2 之 STI 電路的 RTL simulation。

A 等級之 Cell Area 可利用以下指令產生而得知

QoR Report：STI_DAC.qor

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

以 RC 產生 QoR report 的指令 ： ***report qor > STI_DAC.qor***

```
Area
--------------------------------------
Combinational Area:      4255.381786
Noncombinational Area:   5808.502939
Buf/Inv Area:             599.182190
Total Buffer Area:            251.22
Total Inverter Area:          347.97
Macro/Black Box Area:       0.000000
Net Area:               85960.430054
--------------------------------------
Cell Area:              10063.884725
Design Area:            96024.314779
```

C 等級至 D 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 v2012.12 moldelsim v10.2 vcs v2009.09-SP1
Logic Synthesizer	design-compiler v2013.03-sp5

附錄 B 設計檔案說明

1. 下表五.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
testfixture1.v	測試樣本檔 1。此測試樣本檔定義了時脈週期與測試樣本之輸入信號。
testfixture2.v	測試樣本檔 2。此測試樣本檔定義了時脈週期與測試樣本之輸入信號。
STI_DAC.v (STI_DAC .vhd)	參賽者所使用的設計檔，已包含系統輸/出入埠之宣告
synopsys_dc.setup	Design Compiler 初始設定範例檔案
STI_DAC.sdc	Design Compiler 電路合成規範檔
tsmc13_neg.v	Gate-level simulation 所需要之 cell library file
./dat/Pattern.dat	測試樣本 1 之並列資料輸入檔案
./dat/Stimulus.dat	測試樣本 1 之並列資料控制檔案
./dat/Expected_so.dat	測試樣本 1 之序列輸出資料比對檔
./dat/Expected_odd_1.dat ./dat/Expected_odd_2.dat ./dat/Expected_odd_3.dat ./dat/Expected_odd_4.dat	測試樣本 1 之 OM 記憶體 1~4 資料比對檔案
./dat/Expected_even_1.dat ./dat/Expected_even_2.dat ./dat/Expected_even_3.dat ./dat/Expected_even_4.dat	測試樣本 1 之 EM 記憶體 1~4 資料比對檔案
report.000	結果報告範本
./dat/Pattern2.dat	測試樣本 2 之並列資料輸入檔案
./dat/Stimulus2.dat	測試樣本 2 之並列資料控制檔案
./dat/Expected2_so.dat	測試樣本 2 之序列輸出資料比對檔
./dat/Expected2_odd_1.dat ./dat/Expected2_odd_2.dat ./dat/Expected2_odd_3.dat ./dat/Expected2_odd_4.dat	測試樣本 2 之 OM 記憶體 1~4 資料比對檔案
./dat/Expected2_even_1.dat ./dat/Expected2_even_2.dat ./dat/Expected2_even_3.dat ./dat/Expected2_even_4.dat	測試樣本 2 之 EM 記憶體 1~4 資料比對檔案

2. 請使用 ***STI_DAC.v(.vhd)***，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 STI_DAC(clk ,reset, load, pi_data, pi_length, pi_fill, pi_msb, pi_low, pi_end,
               so_data, so_valid,
               oem_finish, oem_dataout, oem_addr,
               odd1_wr, odd2_wr, odd3_wr, odd4_wr, even1_wr, even2_wr, even3_wr, even4_wr);

    input               clk, reset;
    input               load, pi_msb, pi_low, pi_end, pi_fill;
    input       [15:0]  pi_data;
    input       [1:0]   pi_length;
    output              so_data, so_valid;
    output              oem_finish, odd1_wr, odd2_wr, odd3_wr, odd4_wr,
                        even1_wr, even2_wr, even3_wr, even4_wr;
    output      [4:0]   oem_addr;
    output      [7:0]   oem_dataout;
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 STI_DAC.v
   - 若使用 modelsim，則是在 compiler verilog 時，使用下面指令：
     vlog testfixture.v STI_DAC.v
   - 若使用 vcs，則是在 compiler verilog 時，使用下面指令：
     vcs –R +v2k testfixture.v STI_DAC.v
   - 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 STI_DAC.v +notimingchecks***
     vlog testfixture.v STI_DAC.v +notimingchecks
     vcs –R +v2k testfixture.v STI_DAC.v +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 STI_DAC.v +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 STI_DAC.v –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；(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		
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.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

FTP 帳號(FTP account): 999999

通過 gate-level simulation 之 cell area report：20000

--- RTL category---
使用之 HDL 模擬器名稱(HDL simulator): nc-verilog
RTL 檔案名稱(RTL filename): STI_DAC.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): STI_DAC_syn.vg
gate-level sdf filename: STI_DAC_syn.sdf
design compiler 合成資料庫(dc library): STI_DAC_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***。

本題限制上傳之設計檔僅可使用 tar 或 zip 壓縮格式，使用 rar 或其他格式者一律不予計分。

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。