

88 學年度大學校院積體電路設計比賽

研究所 Cell-based 組競賽試題

(for VERILOG)

一、試題說明：

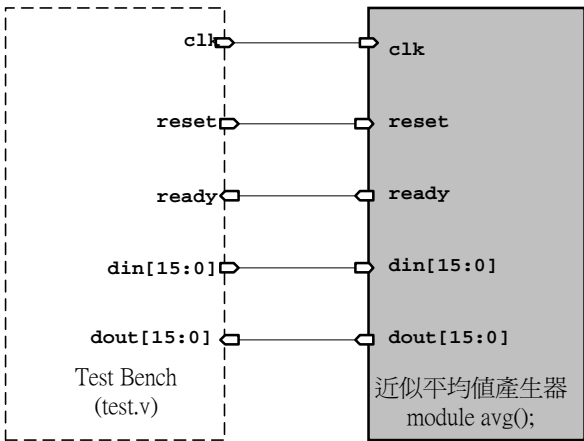
請以 CIC 所提供的 TSMC 1P4M 0.35 μ m Standard Cell library (cb35os142)，設計一個近似平均值產生器模組(不含 I/O PAD)，其功能及規格將詳述於後。參賽者在規定時間內須完成 Front-end 及 Back-end Design，並繳交各項設計相關資料以供主辦單位評分。

1-1 功能描述

設計一近似平均值產生器模組，其 module name 為 avg，如圖一所示；test bench 部分則由 CIC 提供。此模組的輸入資料(din)為連續不斷的 16-bit 正整數，而輸出(dout)則為輸入數字串的近似平均值。假設系統內可儲存 n 個數字，將這 n 個數字相加之後再除以 n 所得到的**商數**為平均值。而本系統所需要輸出的近似平均值則是模組中現有的這 n 個輸入數字中與平均值最接近者。若現有的 n 個輸入數字中有兩個以上的數字與平均值的差值(絕對值)相同，則取其中較小的數字作為輸出的近似平均值。舉例而言，若 n 為 3，且系統內儲存的數字串為 10、5、23，則將(10+5+23)除以 3 得到商數為 12，故其平均值為 12，與 12 最接近的數字是 10，因此要輸出的近似平均值就是 10。另外，若 n 為 4，數字串為 10、5、23、18，此時平均值應為 (10+5+23+18)/4 的商數 14，而與 14 最接近的數字有兩個，分別是 10 和 18，取較小者，故要輸出的近似平均值就是 10。

在本模組中，可儲存 12 個輸入數字，即 n 固定為 12。當 reset 訊號由 0 變成 1 之後，輸出訊號 ready 此時必須被重設為 0。而 reset 由 1 變為 0 之後，第一個輸入數字即已出現在 din 埠上，此後在每個 clock 的上昇緣都會有新的數字資料出現，亦即每個 clock cycle 會由 din 輸入一個新的 16-bit 正整數。一開始，輸入的數字資料未滿 12 個之前，並不需要計算其近似平均值後。當輸入數字的數目 n 等於 12 時，則開始由此 12 個數字決定其近似平均值後輸出，然後將最先讀入系統

的一個數字從系統儲存的輸入數字中移去(First In First Out)。待下一個數字輸入時，加入此新讀入的數字，由此時的 12 個數字決定新的近似平均值後輸出。同樣要再移去此時 12 個數字中最先讀入系統的一個數字。如此一直維持系統中儲存數字的數目為 12，並不斷的產生新的近似平均值。由於每個 clock cycle 會輸入一個數字，因此對輸出而言，在輸出第一個近似平均值之後，同樣也是要每個 clock cycle 輸出一個近似平均值數字。當第一個近似平均值輸出之前，ready 訊號必須由 0 變成 1 以表示下一個 clock cycle 會有第一個近似平均值輸出。附件一是一個本模組的運算範例，供參賽者參考。data path 的設計可善用 SYNOPSIS designware 提供的功能。



腳位名稱		描述	Drive Strength/Output Load
clk	輸入	系統時脈	assume infinite
reset	輸入	系統重置訊號，high active	1 ns/pf
din	輸入	每個clock cycle輸入一個16-bit 正整數	1 ns/pf
ready	輸出	reset為1時，ready=0。第一個近似平均值輸出前的半個cycle，ready由0變成1。	1pf
dout	輸出	每個clock cycle輸出一個16-bit 近似平均值	1pf

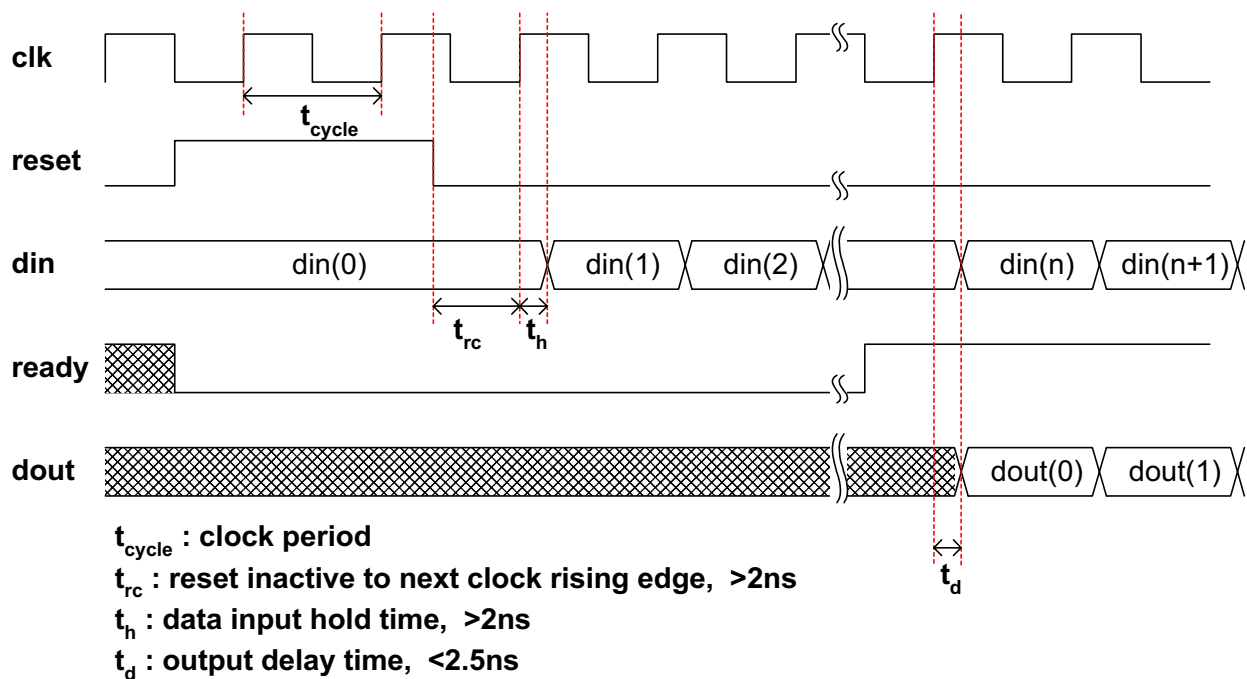
圖一、系統描述

1-2 時序規格

本系統的時序規格如圖二所示，由設計者自行決定時脈週期 t_{cycle} 。reset 訊號為一非同步重置訊號，當 reset 為 1 時表示系統重置，此時 ready 訊號必須被設定為 0，reset 為 1 的時間至少會大於一個 clock cycle，且 reset 訊號由 1 變為 0 後，到下一個 clk 的上昇緣之間的時間必須大於 t_{rc} 。第一個輸入數字，din(0)，在 reset 訊號變為 1 之前已經 valid。當 reset 訊號由 1 變為 0 後下一個 clk 的上昇緣，輸入數字經過 t_{h} 之後變成第二個資料，din(1)。此後每個 clk 的上昇緣經過 t_{h} 之後都會

有資料的轉換。

當近似平均值產生器接受輸入字串之後便開始計算第一個近似平均值， $dout_0$ 。當第一個近似平均值要輸出之前半個 clock cycle， $ready$ 訊號必須由 0 變成 1，以告知下一個 clk 的上昇緣將會有第一個近似平均值輸出。 clk 訊號上升後到正確的近似平均值輸出之間的 delay time 必須小於 t_d 。此後每個 clk 的上升緣都要有一個近似平均值輸出。



圖二、系統時序圖

二 軟體使用注意事項

主辦單位提供以下檔案：

00.README	說明檔
avg.v	近似平均值產生器模組之宣告
test.v	testfixture file
avg_setup.dc	SYNOPSYS 環境設定檔
IN.DAT	input and result data

- ❖ 請用avg.v中的宣稱及template module來建立近似平均值產生器模組。

其 Verilog I/O port name 及 Top module name 的定義如下：

```
module avg(din, reset, clk, ready, dout);
    input reset, clk;
    input [15:0] din;
    output ready;
    output [15:0] dout;
endmodule
```

- ❖ 請使用CIC提供的test bench(**test.v**)進行verilog simulation。評審亦以此test bench為審查主要依據(除此test bench外，評審尚會以其他的test pattern測試參賽者的設計)。
 - test.v包含了Fig.1中虛線部分的模組。test.v會讀入IN.DAT的data並產生clock、reset與din等訊號。等到收到ready訊號，test.v會將dout訊號的輸出資料與IN.DAT中的expect result做比較。若結果正確，則會顯示PASS訊息。若有錯誤，則會顯示發生錯誤的data。
 - 若要改變clock period，可以修改test.v中的"HALF_CYCLE"定義，**clock period = HALF_CYCLE * 2**。（HALF_CYCLE 預設值為10ns，即clock period為20ns）
 - 執行gate-level模擬時，必須使用cb35os142.v搭配SYNOPSYS以及avg_setup.dc產生的sdf檔案。

- ❖ avg_setup.dc 檔案提供了本模組在 SYNOPSYS 中的基本設定，包括：

- 工作環境為 NCCOM。
- 除 clk 外，所有 input port 的 driving strength 為 1 ns/pf。
- 所有 output port 的 loading 為 1pf。

請用 Setup/Execute Script 將此檔案載入。並依此環境設定產生 gate-level netlist 的 sdf 檔案以配合 verilog 之 gate-level simulation。若參賽者不經由 SYNOPSYS

產生 gate-level netlist，也必須將其 gate-level netlist 讀入 SYNOPSIS 中以同樣 avg_setup.dc 的設定產生 sdf 檔案。

- ❖ 限定使用 block layout 的方式 (即不含 I/O pad，只含 core cell 的方式)。
 - 在 Physical Design 做 routing 前請將 block 的 i/o pin 調到 grid 上。(在 Place&Route 選單=>Block=>Adjust Pins, 按 OK)。
 - Power Ring 及 Power Pin 的寬度：vdd! 和 gnd! 各 40um。
 - 一律不加 power strip，也不做 clock tree synthesis，也不加 core filler。
 - Signal Pin 的位置由參賽者自行決定。
 - 其餘有關 block layout 應注意的事項可參閱下列網站 http://www.cic.edu.tw/~chtsai/technique_doc/note_block_layout.htm 的說明。
- ❖ 佈局驗證(DRC, ERC 及 LVS)請使用 Dracula(Cadence 97A)。跑 Dracula 需在 layout 加上 text label,請使用 layout editor 以(text2 drawing) layer 當作 text 打在 i/o pin 的位置。
- ❖ 為避免主辦單位驗證結果時的困擾，所有電路之輸出、輸入及電源等接腳須與題目指定之名稱相同。

三、繳交資料

1. Verilog RTL-level synthesizable code 檔案。
2. Verilog Gate-level netlist 檔案。
3. 完整的設計資料庫(Cadence design library)
請利用 tar 整合成一個 .tar 檔，假設你的 library name 為 **your_lib**。經由 tar 指令

```
>tar cvf yourname.tar your_lib
```

將會得到 yourname.tar 檔名的檔案。

4. 佈局之 GDSII 檔案及 DRC 與 LVS 驗證檔案(*.sum 及 *.lvs 檔)。
 5. 如欲說明電路架構及特殊創意之處可另以文字或圖片加以說明。(以 MS-word 檔案或純文字檔描述)
- ❖ 請另建一個新目錄，並將以上各項需要繳交的檔案複製到此新目錄下。在此目錄下執行以下指令將所有檔案整合並壓縮為一個檔案：

```
> tar cvf xxxxxxxx.tar *
```

↑

檔案名稱自取

```
> compress xxxxxxxx.tar
```

經由以上指令可得到 xxxxxxxx.tar.Z 的檔案。

- ❖ 文件說明檔(report.xxx):

本項檔案係用來說明參賽者繳交的各相關檔案之檔案名稱、使用軟體項目、相關規格及其他說明事項。本項檔案格式由主辦單位提供，請參照隨題目所取回的檔案中的 **report.000** 檔，將相關資料名稱填入檔案中。

- ❖ 請將最後壓縮整合的檔案(xxxxxxx.tar.Z)以及文件說明檔(report.xxx)使用 binary 模式利用 **icresult** 的帳號及密碼傳送至以下四個傳送網站之一即可。(請先上傳 xxxxxxxx.tar.Z，再上傳 report.xxx)

台灣大學：video3.ee.ntu.edu.tw(140.112.17.191)

晶片中心：dc89.cic.edu.tw(140.126.26.43)

雲林科技大學：cad.el.yuntech.edu.tw(140.125.35.1)

成功大學：cad9.ee.ncku.edu.tw(140.116.156.159)

傳送的目錄為 inst_cell/參賽隊號。

- ❖ 各項設計資料檔如需更新時，請重複以上步驟，並另取新的檔名傳送，注意務必更改文件說明檔 **report.xxx** 的相關內容。
- ❖ 文件說明檔的檔名須以 **report** 為檔名開頭，副檔名請以數字依序命名，如

report.000 代表原始檔名，**report.001** 代表第一個更新版本，如另有更新，請依此類推。

- ✦ 其他相關事項請參考參賽手冊。

四、評分方式：

1. 經主辦單位驗證其電路功能正確及各項資料完備者始予以完整計分。資料不完備者或設計不完整者，評審委員會依照其設計完成程度酌予給分。

2. 分數計算公式為：

$$\begin{aligned} &\text{if } (Time \geq 10ns) \text{ then} \\ &\quad Score = Area \times (Time/10ns)^2 \\ &\text{else} \\ &\quad Score = Area \times \sqrt{(Time/10ns)} \end{aligned}$$

Score 最低者為第一名。若有 Score 相同及接近者，由評審委員依照各設計之創意排定名次。

❖ Time：即 clock period。

❖ Area：僅計算 design boundary 的面積，不含 IO pad。在 Silicon Ensemble 視窗中，圍繞在晶片周圍的一個紫色框即為 design boundary。

附件一:範例

假設輸入數字串為 66, 22, 39, 68, 122, 22, 90, 29, 63, 16, 10, 50, 35, 47, 126, 69.....。

1. INPUT NUMBER = 66
STORE 66 ==> STRING : 66
2. INPUT NUMBER = 22
STORE 22 ==> STRING : 66, 22
3. INPUT NUMBER = 39
STORE 39 ==> STRING : 66, 22, 39
4. INPUT NUMBER = 68
STORE 68 ==> STRING : 66, 22, 39, 68
5. INPUT NUMBER = 122
STORE 122 ==> STRING : 66, 22, 39, 68, 122
6. INPUT NUMBER = 22
STORE 22 ==> STRING : 66, 22, 39, 68, 122, 22
7. INPUT NUMBER = 90
STORE 90 ==> STRING : 66, 22, 39, 68, 122, 22, 90
8. INPUT NUMBER = 29
STORE 29 ==> STRING : 66, 22, 39, 68, 122, 22, 90, 29
9. INPUT NUMBER = 63
STORE 63 ==> STRING : 66, 22, 39, 68, 122, 22, 90, 29, 63
10. INPUT NUMBER = 16
STORE 16 ==> STRING : 66, 22, 39, 68, 122, 22, 90, 29, 63, 16
11. INPUT NUMBER = 10
STORE 10 ==> STRING : 66, 22, 39, 68, 122, 22, 90, 29, 63, 16, 10
12. INPUT NUMBER = 50
STORE 50 ==> STRING : 66, 22, 39, 68, 122, 22, 90, 29, 63, 16, 10, 50
AVERAGE = $597/12 = 49$
OUTPUT = 50
REMOVE 66
13. INPUT NUMBER = 35
STORE 35 ==> STRING : 22, 39, 68, 122, 22, 90, 29, 63, 16, 10, 50, 35
AVERAGE = $566/12 = 47$
OUTPUT = 50
REMOVE 22
14. INPUT NUMBER = 47
STORE 47 ==> STRING : 39, 68, 122, 22, 90, 29, 63, 16, 10, 50, 35, 47
AVERAGE = $591/12 = 49$
OUTPUT = 50
REMOVE 39
15. INPUT NUMBER = 126
STORE 126 ==> STRING : 68, 122, 22, 90, 29, 63, 16, 10, 50, 35, 47, 126
AVERAGE = $678/12 = 56$
OUTPUT = 50
REMOVE 68

16. INPUT NUMBER = 69

STORE 69 ==> STRING :122, 22, 90, 29, 63, 16, 10, 50, 35, 47, 126, 69

AVERAGE = $679/12 = 56$

OUTPUT = 50

REMOVE 122

17. INPUT NUMBER = 98

STORE 98 ==> STRING : 22, 90, 29, 63, 16, 10, 50, 35, 47, 126, 69, 98

AVERAGE = $655/12 = 54$

OUTPUT = 50

REMOVE 22

18. INPUT NUMBER = 83

STORE 83 ==> STRING : 90, 29, 63, 16, 10, 50, 35, 47, 126, 69, 98, 83

AVERAGE = $716/12 = 59$

OUTPUT = 63

REMOVE 90

19. INPUT NUMBER = 98

STORE 98 ==> STRING : 29, 63, 16, 10, 50, 35, 47, 126, 69, 98, 83, 98

AVERAGE = $724/12 = 60$

OUTPUT = 63

REMOVE 29