

89 學年度大專校院積體電路設計競賽試題

一、試題說明:

請設計一個 Adaptive Differential PCM 編解碼電路(ADPCM Encoder/Decoder)，此一電路包含一個可將 16-bit 訊號壓縮為 4-bit 編碼(Code)之編碼器(Encoder)，以及一個可將 4-bit 編碼還原為 16-bit 訊號之解碼器(Decoder)，此編解碼電路之功能及規格將詳述於 1-1 及 1-2。參賽者在規定時間內須以 CIC 所提供的 Avant! 0.35 μm cell library，及 Test Bench 完成 Front-end 及 Back-end Design 與驗證，並繳交各項設計相關資料以供主辦單位評分(細節詳述於第三節-繳交檔案中)。

1-1 系統描述

假設有一編解碼系統其方塊圖如 Figure 1.所示，此系統中的兩個模組功能簡述如下:

Module Name	Description
Test Bench	Test Module, given
CODEC	CODER/DECODER Module

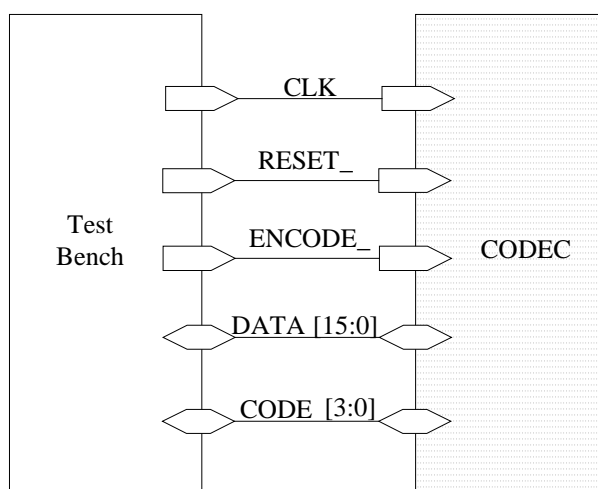


Figure 1. 資料編解碼系統

Figure 2. 描述這個編解碼系統的時序圖，Test Bench 會產生 CLK 訊號給 CODEC 電路，在資料處理前，Test Bench 會設定 ENCODE_ 訊號決定讓 CODEC module 執行編碼或解碼(Low 為編碼, High 為解碼)的動作，經過一個 CLOCK 週期之後，Test Bench 產生一個 Low Active RESET_ 訊號，CODEC module 便將內部的暫存器設定初始值，之後，CODEC module 便開始依據 ENCODE_ 訊號做編碼或解碼的運作：若為編碼運算則 CODEC module 須設定 DATA 成為 input port，CODE 成為 output port，依序在每一個 CLK 上升緣從 DATA 讀入 Test Bench 產生的一個 16-bit 的資料，然後壓縮編碼成 4-bit 資料，再將編碼結果由 CODE 輸出，Test Bench 此時會於每一個 CLK 正緣時，將結果讀入做比對。若為解碼動作，則 CODEC

module 須設定 DATA 成為 output port，CODE 成為 input port，並於接到 RESET_ 訊號變為 Low 之後，由 CODE 依序讀入一個 4-bit 的資料，然後解壓縮還原成 16-bit 資料，再將解碼結果由 DATA 輸出 Test Bench 做驗證。

CODEC 會持續編碼動作或解碼動作直到 Test Bench 重新設定 ENCODE_ 或 RESET_ 為止。

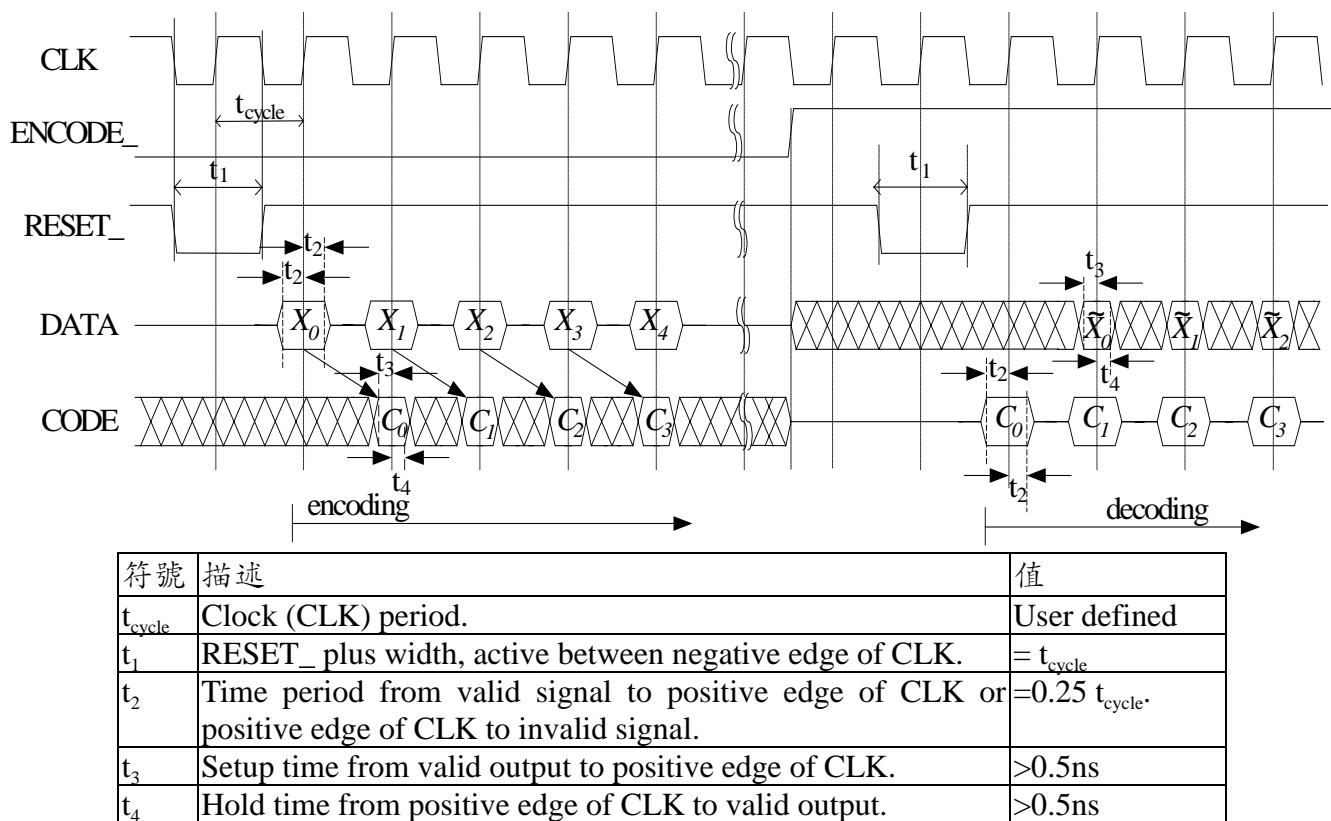


Figure 2. 系統時序圖

1-2 功能說明

本電路之編碼器即解碼器架構如 Figure 3 及 Figure 4 所示此，為簡化起見，其參數說明如下：

X_n : 16-bit 輸入訊號(2's complement 整數)，於 CLK 正緣時，由 DATA 取得的值。

\tilde{X}_n : 16-bit 合成訊號(2's complement 整數)， $\tilde{X}_n = \tilde{d}_n + \tilde{X}_{n-1}$ ，初始值=0，如果 \tilde{X}_n 發

生 overflow，即當 $\tilde{X}_n > 16'h7fff$ (32767)則 $\tilde{X}_n = 16'h7fff$ 或當 $\tilde{X}_n < 16'h8000$ (-

32768)則 $\tilde{X}_n = 16'h8000$ 。

d_n : 13-bit 整數誤差訊號(differential signal) (2's complement 整數)， $d_n = X_n - \tilde{X}_{n-1}$ ，

若產生 overflow，即如 $X_n - \tilde{X}_{n-1} \geq 2^{12}$ 則 $d_n = 13'b01111111111111$ (4095)，如

$X_n - \tilde{X}_{n-1} < -2^{12}$ 則 $d_n = 13'b10000000000000$ (-4096),。

Δ_n : 9-bit 正整數，下一筆的量化步距，每個週期依據所得到的 C_n ，計算下一筆的量化步距： $\Delta_n = \Delta_{n-1} * M(C_n)$ ，初始值 = 1，若產生 overflow，即當 $(\Delta_{n-1} * M(C_n)) \geq 2^9$ ，則 $\Delta_n = 9'b11111111$ (511)，若發生 underflow，即當 $(\Delta_{n-1} * M(C_n)) < 1$ ，則 Δ_n 設成最小值($\Delta_n=1$)。而 $\Delta_{n-1} * M(C_n)$ 的小數部分採用無條件消去法，讓 Δ_n 成為 9-bit 正整數。

C_n : 4-bit 編碼訊號。

\tilde{d}_n : 13-bit 2's complement 整數，以 Δ_{n-1} 量化所計算出的誤差訊號(d_n)即以 Δ_{n-1} 為單位算出最接近 d_n 的整數倍。

Z^{-1} : Unit delay (暫存器)。

編碼步驟：將輸入訊號 X_n 與上一次合成的訊號 \tilde{X}_{n-1} 相減求出誤差訊號 d_n ，然後再將 d_n 依照上一次計算出的 step size 值 Δ_{n-1} 與 Figure 5. 的量化曲線對應，其他相關數值對應表描述於 Table 1. 中。編碼壓縮成 C_n 輸出。最後再根據編碼結果 C_n 與 Table 1. 步距函數對應，計算出新的 Δ_n 值供下一次運算使用。

解碼步驟：輸入編碼訊號 C_n ，對應上一次計算出的 Δ_{n-1} 與 Table 2. 的反量化表，計算出量化誤差訊號 \tilde{d}_n 後，再加上上一次合成的訊號 \tilde{X}_{n-1} ，便可以解碼還原出這一次新的訊號預估值 \tilde{X}_n 輸出。同時，根據輸入訊號 C_n 與 Table 1. 步距函數對應，計算出新的 Δ_n 值供下一次運算使用。

於附表一描述一個範例說明編解碼的動作方式與產生的數值。

注意：此編解碼器的編碼和解碼動作不會同時進行。

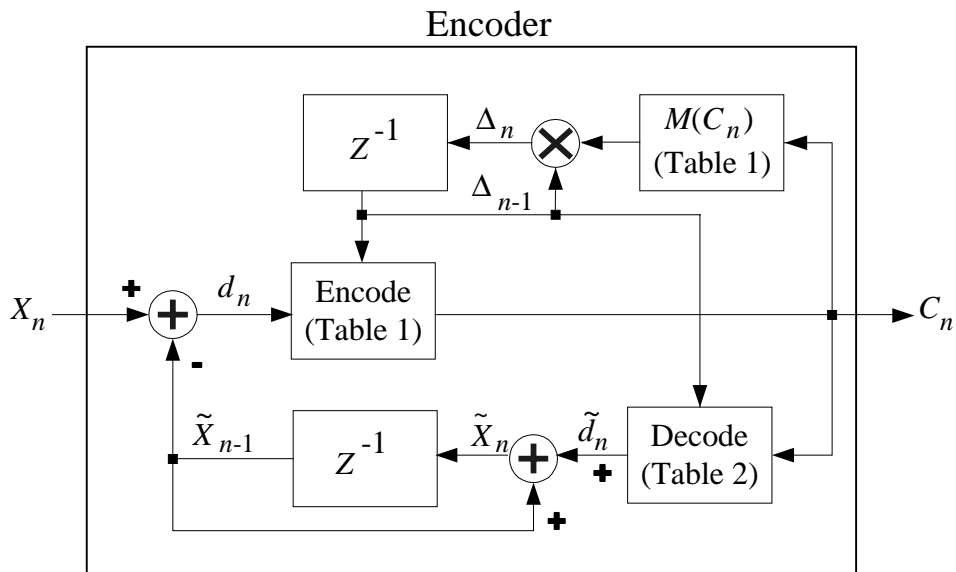


Figure 3. 編碼器架構

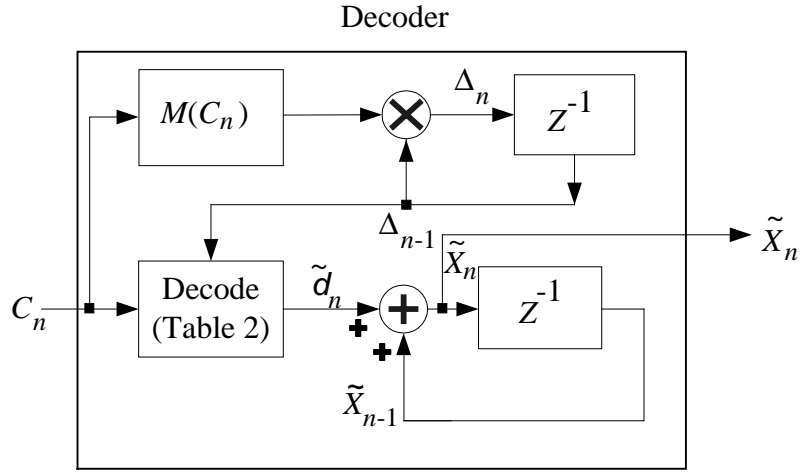


Figure 4. 解碼器架構

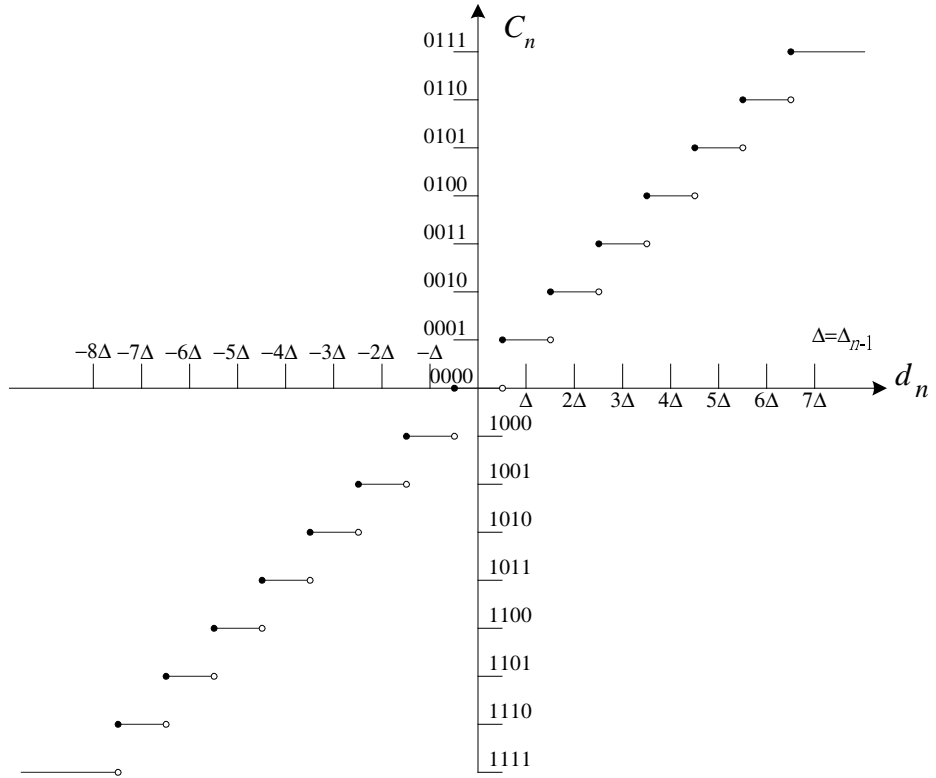


Figure 5. 編碼量化器曲線對應表

d_n 範圍	Cn	$M(Cn)$	\tilde{d}_n	Cn	\tilde{d}_n
$d_n < -\frac{15}{2}\Delta$	1111	2.375	-8Δ	1111	-8Δ
$-\frac{15}{2}\Delta \leq d_n < -\frac{13}{2}\Delta$	1110	2.375	-7Δ	1110	-7Δ
$-\frac{13}{2}\Delta \leq d_n < -\frac{11}{2}\Delta$	1101	2.0	-6Δ	1101	-6Δ
$-\frac{11}{2}\Delta \leq d_n < -\frac{9}{2}\Delta$	1100	1.625	-5Δ	1100	-5Δ
$-\frac{9}{2}\Delta \leq d_n < -\frac{7}{2}\Delta$	1011	1.125	-4Δ	1011	-4Δ
$-\frac{7}{2}\Delta \leq d_n < -\frac{5}{2}\Delta$	1010	0.875	-3Δ	1010	-3Δ
$-\frac{5}{2}\Delta \leq d_n < -\frac{3}{2}\Delta$	1001	0.875	-2Δ	1001	-2Δ
$-\frac{3}{2}\Delta \leq d_n < -\frac{1}{2}\Delta$	1000	0.875	$-\Delta$	1000	$-\Delta$
$-\frac{1}{2}\Delta \leq d_n < \frac{1}{2}\Delta$	0000	0.875	0	0000	0
$\frac{1}{2}\Delta \leq d_n < \frac{3}{2}\Delta$	0001	0.875	Δ	0001	Δ
$\frac{3}{2}\Delta \leq d_n < \frac{5}{2}\Delta$	0010	0.875	2Δ	0010	2Δ
$\frac{5}{2}\Delta \leq d_n < \frac{7}{2}\Delta$	0011	0.875	3Δ	0011	3Δ
$\frac{7}{2}\Delta \leq d_n < \frac{9}{2}\Delta$	0100	1.125	4Δ	0100	4Δ
$\frac{9}{2}\Delta \leq d_n < \frac{11}{2}\Delta$	0101	1.625	5Δ	0101	5Δ
$\frac{11}{2}\Delta \leq d_n < \frac{13}{2}\Delta$	0110	2.0	6Δ	0110	6Δ
$\frac{13}{2}\Delta \leq d_n$	0111	2.375	7Δ	0111	7Δ

Table 1. 編碼量化器與步距調整函數對應表

Table 2. 解碼反量化器對應表

二、軟體使用注意事項:

2-1 主辦單位提供以下檔案:

00.README: 說明檔

testfixture.v: 測試檔，含定義 clock 週期與測試值輸入檔。

codec.v: 輸入輸出宣告。

synopsys.dc: Synopsys 設定 operating conditions 和 boundary conditions 檔(不含 design constraint 部分)。

INa.DAT: 4096 筆一般語音的輸入資料以及預期結果檔。

INb.DAT: 4096 筆由正弦波、鋸齒波、方波、和隨機的亂數組成的輸入資料以及預期結果檔。

report.000: 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。

2-2 請利用 codec.v 內的 module codec 來建立編解碼器，其 IO port 的定義及模組名稱的宣告如下：

```
module CODEC (DATA, CODE, RESET_, ENCODE_, CLK);
```

```
input CLK, ENCODE_, RESET_;
```

```
inout [15:0] DATA;
```

```
inout [3:0] CODE;
```

```
endmodule
```

2-3 請利用 CIC 提供的 test bench(testfixture.v)來驗證設計的正確性，評審會以此 test bench 來驗證參賽者設計的正確性(另外評審尚會以其他的 test pattern 來驗證參賽者的設計)。

- i. testfixture.v 中包含 Figure.1. 的 Test Bench，一開始 Test Bench 會產生 CLK, CODEC_, RESET_,等訊號，驗證編碼動作，首先將 INa.DAT 或 INb.DAT 讀入並每一 CLK 週期將 INa.DAT 或 INb.DAT 一筆一筆的放於 IO port: DATA，並依 Figure. 2 的時序圖，將產生出已編碼的資料從 IO port: CODE 讀入並做驗證，當處理完 4096 筆輸入的資料後，test bench 會重設 CODEC_, RESET_ 訊號驗證解碼的運作，首先將 INa.DAT 或 INb.DAT 中的解碼預期值讀入並每一 CLK 週期將資料一筆一筆的放於 CODE 並依 Figure. 2 的時序圖，將產生出已編碼的資料從 DATA 讀入並做驗證。故驗證時可以更改 testfixture.v 中所定義的 CYCLE 和 INFILE 來分別改變 CLK 週期和 test pattern 檔。
- ii. 編解碼都沒有問題會產生 PASS 的提示，如果和參考值不相符，則會顯示出筆數、錯誤值及參考值。
- iii. synopsys.dc 提供 SYNOPSYS 的基本設定但不包含 design constraints，此 synopsys.dc 只包含：
 1. 工作環境為 WCCON.
 2. 除 CLK 外，所有 input port 和 inout port 的 driving strength 為 1ns/1pf.
 3. 所有的 inout port 的 load 為 1pf.

請利用 Design Analyzer 中的 Setup/Execute Script 來載入 synopsys.dc 或是在 dc_shell 中用 “include synopsys.dc” (dc_shell>include synopsys.dc) 來載入設定檔，並產生 gate-level netlist 和 sdf 檔，搭配 Verilog 之模擬。

- iv. 執行 gate level 模擬時，須搭配 cb35os142.v 和 SYNOPSYS 所產生出的 sdf 檔來模擬。若不是經由 SYNOPSYS 所產生的 gate level netlist 也需要將此 netlist 以 SYNOPSYS 讀入，並配合 synopsys.dc 的設定產生 sdf 檔來配合 Verilog 模擬(細節請參考 CIC cell-based design kit 中 00.Readme.Verilog 的描述)。

- v. 限定產生 block layout 方式 (即不含 IO pad，只含有 core cell 的方式):
 1. power ring 和 power pin 的寬度: vdd! 和 gnd! 皆固定為 20um.
 2. Signal pin 的位置由參賽者自行決定。
 3. 其餘有關 block layout 應注意的事項 Cadence SE 使用者可參閱下列網站:
<http://www.cic.edu.tw/~nschang/seflow/hardmacro/hardmacro.htm> 的說明，
 Avanti Apollo 使用者可參閱下列網站:
http://www.cic.edu.tw/~cschen/design_without_pad.html 的說明。
- vi. 佈局驗證 (DRC 和 LVS):
 - ✧ Cadence SE 使用者：請使用 Dracula，使用 Dracula 作 LVS 需要於 layout 加上 text label，詳細做法請參閱上述網站之介紹。為避免主辦單位驗證結果時的困擾，所有電路的輸出、輸入及電源等接腳名稱須與題目指定之名稱相同。
 - ✧ Avanti Apollo 使用者：請使用 Apollo 內建之 DRC, LVS 驗證之後，再使用 Dracula double check DRC 部分。

三、繳交檔案：參賽隊伍需繳交以下資料：

3-1 設計資料庫

- i. Cadence SE 使用者: Cadence library 完整的設計資料庫(icfb 所建的 library 之目錄)，請利用 tar 整合成一個 .tar file. If your library name is **your_lib**:

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

 You will get *yourname.tar* file.
 Cadence SE 使用者請額外附 SE 所產生的 .def 檔。
- ii. Avanti Apollo 使用者: 將 Apollo 的 library 目錄，利用 tar 整合成一個 .tar file. If your library directory is **your_lib**:

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

 You will get *yourname.tar* file.

3-2 Verilog RTL-level synthesizable code 檔案

3-3 Verilog gate-level netlist file.

3-4 佈局檔 GDSII file。

3-5 佈局驗證輸出檔

- i. Cadence SE 使用者：DRC 結果(*.sum)及 LVS 結果(*.lvs 檔)。
- ii. Avanti Apollo 使用者：Apollo 內建之 DRC/LVS 結果文字檔和 Dracula DRC 之 *.sum 檔。

3-6 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壓縮並整合為一個檔案：

```
> tar cvf xxxxxxxx.tar *
      ↑
      檔案名稱自取
> compress xxxxxxxx.tar
```

經由以上指令可得到 **xxxxxxx.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 **xxxxxxx.tar.Z**，再上傳 **report.xxx**)
 - 台灣大學：video4.ee.ntu.edu.tw(140.112.17.192)
 - 晶片中心：dc90.cic.edu.tw(140.126.24.100)
 - 雲林科技大學：cad15.cad.el.yuntech.edu.tw(140.125.35.15)
 - 成功大學：cad9.ee.ncku.edu.tw(140.116.156.159)傳送的目錄為 **grad_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:

4-1 經主辦單位驗證其電路功能正確及各項資料完備者使予以完整計分，如果資料不完備或設計不完整者，將依照其設計完成程度酌予給分。

4-2 分數計算公式為：

If ($Time \geq 25$) then

$$Score = Area \times \left(\frac{Time}{25}\right)^2$$

else

$$Score = Area \times \sqrt{\frac{Time}{25}}$$

Time: clock period，單位為 ns。

Area: design boundary 的面積，即 Silicon Ensemble 視窗中，圍繞在晶片周圍的紫色框的面積或是 Apollo 的黃色虛線框的面積。

Score 最低者為第一名，若有分數相同者幾接近者，由評審委員依照各設計之創意排定名次。

附件一、範例:

底下為一範例之時序表,全部採用 16 進位表示：

時序	Encoding								Decoding				
	X_n	\tilde{x}_{n-1}	d_n	Δ_{n-1}	C_n	\tilde{d}_n	\tilde{x}_n	Δ_n	C_n	\tilde{d}_n	\tilde{x}_{n-1}	Δ_{n-1}	\tilde{X}_n
0	0	0	0	1	0	0	0	1	0	0	0	1	0
1	39	0	39	1	7	7	7	2	7	7	0	1	7
2	80	7	79	2	7	e	15	4	7	e	7	2	15
3	9a	15	85	4	7	1c	31	9	7	1c	15	4	31
4	8c	31	5b	9	7	3f	70	15	7	3f	31	9	70
5	61	70	fff1	15	8	1feb	5b	12	8	1feb	70	15	5b
6	4d	5b	fff2	12	8	1fee	49	f	8	1fee	5b	12	49
7	37	49	ffee	f	8	1ff1	3a	d	8	1ff1	49	f	3a
8	46	3a	c	d	1	d	47	b	1	d	3a	d	47
9	61	47	1a	b	2	16	5d	9	2	16	47	b	5d
a	1f	5d	ffc2	9	e	1fc1	1e	15	e	1fc1	5d	9	1e
b	ffd0	1e	ffb2	15	b	1fac	ffca	17	b	1fac	1e	15	ffca
c	ff81	ffca	ffb7	17	a	1fbb	ff85	14	a	1fbb	ffca	17	ff85
d	ff81	ff85	fffc	14	0	0	ff85	11	0	0	ff85	14	ff85
e	ffad	ff85	28	11	2	22	ffa7	e	2	22	ff85	11	ffa7
f	ffcd	ffa7	26	e	3	2a	ffd1	c	3	2a	ffa7	e	ffd1
10	f	ffd1	3e	c	5	3c	d	13	5	3c	ffd1	c	d
11	46	d	39	13	3	39	46	10	3	39	d	13	46
12	67	46	21	10	2	20	66	e	2	20	46	10	66
13	82	66	1c	e	2	1c	82	c	2	1c	66	e	82
14	a9	82	27	c	3	24	a6	a	3	24	82	c	a6
15	ac	a6	6	a	1	a	b0	8	1	a	a6	a	b0
16	98	b0	ffe8	8	a	1fe8	98	7	a	1fe8	b0	8	98
17	a5	98	d	7	2	e	a6	6	2	e	98	7	a6
18	a4	a6	fffe	6	0	0	a6	5	0	0	a6	6	a6
19	8c	a6	ffe6	5	c	1fe7	8d	8	c	1fe7	a6	5	8d
1a	8c	8d	ffff	8	0	0	8d	7	0	0	8d	8	8d

在時序 1 時，輸入 X_n 為 0x39, delay element 的輸出 \tilde{x}_{n-1} 為 0, 故 d_n 的結果為 0x39，此時 Δ_{n-1} 為 0x1，經過 Table 2 的對應表得 C_n 為 0x7，並更新 delay element 的值: $\tilde{x}_n=0x7$; $\Delta_n=0x2$ ，在時序 2 時，輸入 X_n 為 0x80, delay element 的輸出 \tilde{x}_{n-1} 為時序 0 所更新的值: 0x7, 故 d_n 的結果為 0x79，此時 Δ_{n-1} 為時序 0 更新的值: 0x2，經過 Table 2 的對應表得 C_n 為 0x7，並更新 delay element 的值: $\tilde{x}_n=0x15$; $\Delta_n=0x4$ ，依此方式運作。