

2019 IC Design Contest Final

大學類標準元件數位電路設計

Real-Time GPS Distance Calculator

前言

在行動裝置與物聯網盛行的現代社會，GPS(Global Positioning System)資料相當容易取得，因此也被廣泛的應用於行為分析的研究，例如：交通狀況分析、野生動物生活圈追蹤、甚至疾病擴散預測…等等。

由於 GPS 資料量一般都相當龐大(Big Data)，當使用軟體進行分析時，通常會耗費許多時間用於計算 GPS 訊號點彼此間的距離關係。然而，在有即時性分析(Real-Time Analysis)需求的應用上，例如及時預測，要處理龐大的 GPS 資訊時，此一系統架構往往會犧牲部份資料(精準度降低)，降低需要分析的資料量以滿足時間上的要求，為了解決這個問題，本題的目標是實現一硬體用以計算出目前輸入的 GPS 點與上一次輸入的 GPS 點間的距離資訊，使此資訊能夠直接在軟體端使用，以降低軟體端分析時間。

在計算兩個 GPS 點間的距離時必須考慮到地球的角度問題(球面三角學)，因此衍生出許多不同的方法，運用三角函數之間的關係以計算出球面多邊形的邊和角，包含 Vincenty Formula，Haversine Formula，Great Circle 等，其中 Haversine Formula 的計算複雜度較低且精準度尚可，相當適合大量的 Big Data Analysis，因此，本題目將以硬體實現 Haversine Formula ([請參照 2.3.1](#))。

1. 問題描述

請完成一即時全球定位系統距離運算電路 (Real-Time GPS Distance Calculator, 後文以 GPSDC 表示), 即完成公式一半正矢定理運算式。其電路架構如圖 1 所示, 此電路能夠接收 testbench 發送 GPS 的經、緯度座標資料 (緯度: Latitude; 經度: Longitude), 並能夠利用半正矢公式計算出每兩點經、緯度座標之間的距離。GPSDC 電路在運算時, 將需要使用到 $\cos(\varphi A)$ 、 $\cos(\varphi B)$ 、 $\sin^{-1}(\sqrt{a})$ 的運算, testbench 以查表的方式, 提供定值的運算結果, 參賽者必須利用 **兩點線性內插公式**, 計算出其內插值, 當 GPSDC 運算後輸出中繼運算值 (a)、距離資料 (D) 與運算結果資料有效訊號 (Valid) 給 testbench 進行資料確認, 運算式如 2.3.1 公式 1 所示。

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, TSRI 會根據第 3 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 C 中所列的要求, 附上評分所需要的檔案。

軟體環境及設計資料庫說明請參考附錄 A 與附錄 E。

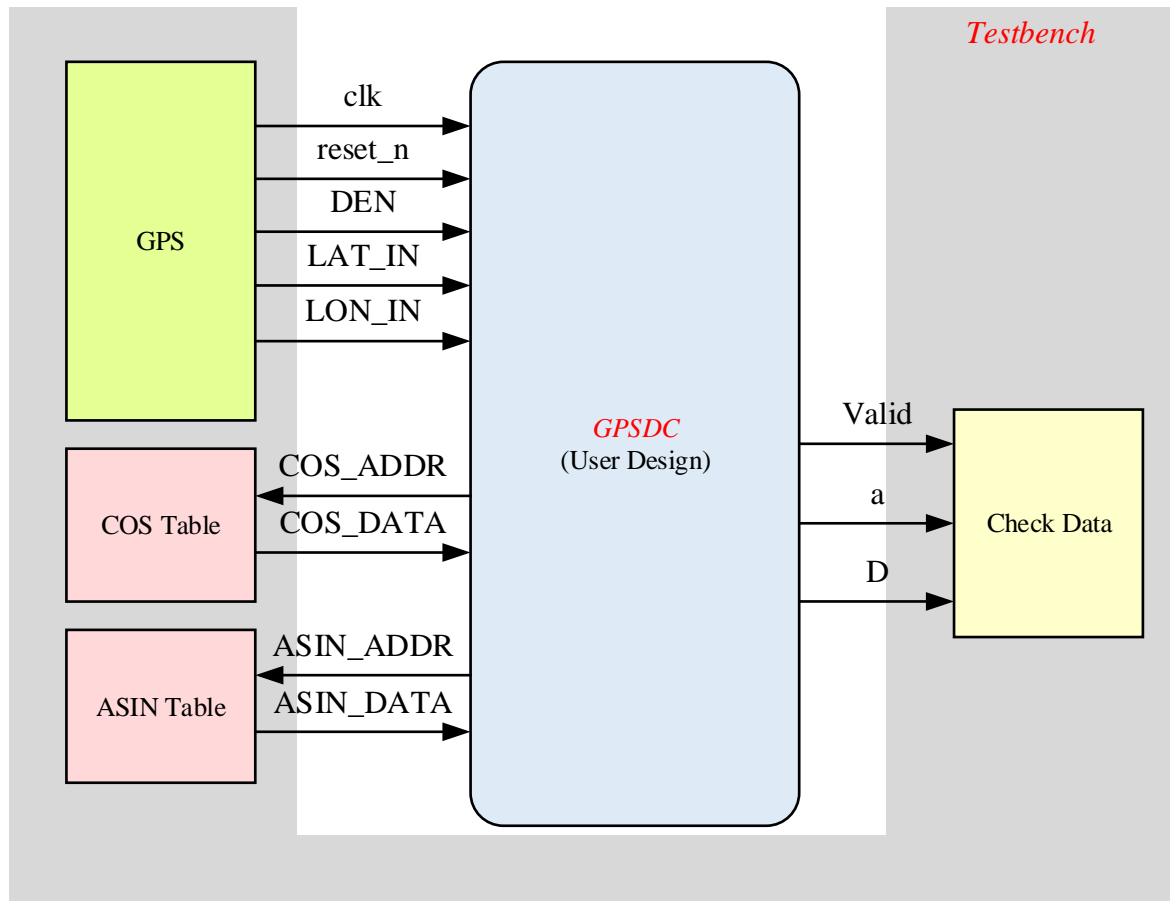


圖 1. Real-Time GPS Distance Calculator 系統架構

本題目之測試樣本置於 `/usr/cad/icc2019/euc/icc2019cb.tar`, 請執行以下指令取得測試樣本:

```
tar xvf /usr/cad/icc2019/euc/icc2019cb.tar
```

2. 設計規格

2.1 系統方塊圖

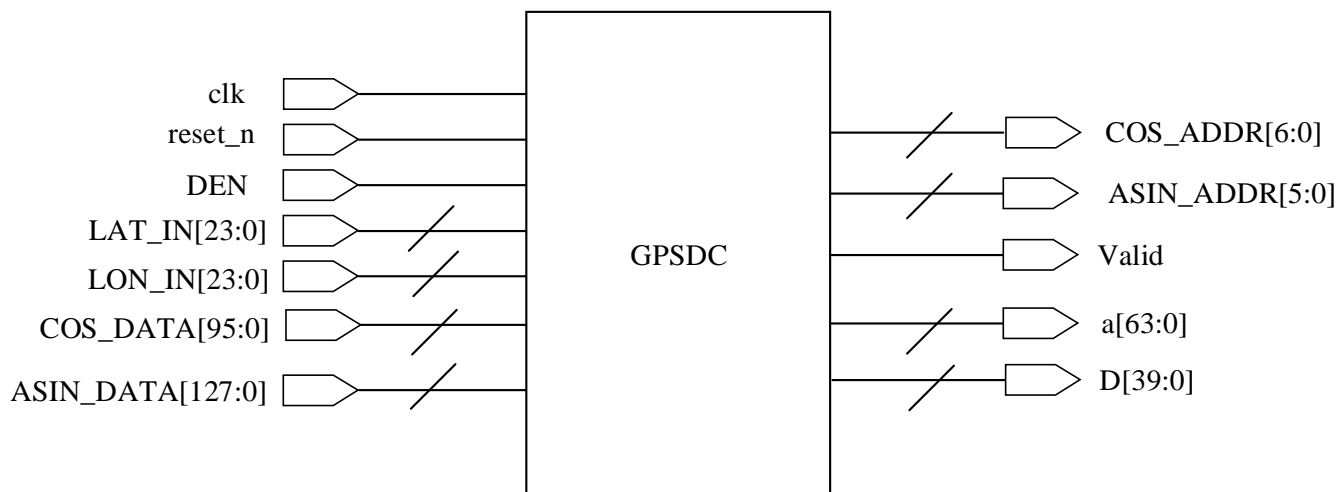


圖 2. 系統方塊圖

2.2 輸出入訊號和記憶體描述

表一、輸入/輸出信號

Signal Name	I/O	Width	Simple Description
clk	I	1	系統時脈訊號。本系統為同步於時脈 正緣 之同步設計。
reset_n	I	1	低位準”非”同步(active low asynchronous)之系統重置信號。
DEN	I	1	GPS 電路資料有效訊號。當此訊號為 High 時，表示 GPS 送出經、緯度訊號為有效資料。 請參照 2.4 說明。
LAT_IN	I	24	GPS 電路緯度座標資料訊號。 此訊號整數為 8 bit，小數為 16 bit。 範例：24'h18C8DD = 24.784628 ₍₁₀₎
LON_IN	I	24	GPS 電路經度座標資料訊號。 此訊號整數為 8 bit，小數為 16 bit。 範例：24'h 79002D = 121.000687 ₍₁₀₎
COS_ADDR	O	7	cos 值查表位址訊號。
COS_DATA	I	96	cos 值資料複合訊號。 此輸出訊號包含 cos 的輸入值及輸出值。 cos 輸入值為 COS_DATA[95:48]，整數部分為 16 bit，小數部分為 32bit 表示。 cos 輸出值為 COS_DATA[47:0]，整數部分為 16 bit，小數部分為 32bit 表示。

ASIN_ADDR	O	6	arcsin 值查表位址訊號。
ASIN_DATA	I	128	arcsin 值資料複合訊號。 此輸出訊號包含 arcsin 的輸入值及輸出值。 arcsin 輸入值為 ASIN_DATA[127:64]，整數部分為 0 bit，小數部分為 64bit 表示。 arcsin 輸出值為 ASIN_DATA[63:0]，整數部分為 0 bit，小數部分為 64bit 表示。
Valid	O	1	距離輸出有效訊號。當此訊號為 High 時，testbench 將會判斷 D 輸出訊號，並送出下一筆經、緯度訊號。 請參照 2.4 說明。
a	O	64	中繼運算值輸出訊號，此訊號整數位為 0 bit，小數位必需精確至 64 bit，小數位第 65 位以後，無條件捨去。
D	O	40	距離計算輸出訊號。 此訊號整數為 8 bit，小數位必需精確至 32 bit，小數位第 33 位以後，無條件捨去。

2.3 系統功能描述

當 reset_n 啟動結束後，testbench 會開始送出資料有效訊號(DEN)、緯度座標訊號(LAT_IN)、經度座標訊號(LON_IN)，參考圖 8 GPSDC 電路初始化輸入之時序規格。當 testbench 送出第二點座標之後，系統開始等待 GPSDC 電路的距離輸出有效訊號(Valid)為 High，才會送出下一組座標資料，同時 Testbench 開始比對中斷運算結果(a)及距離計算訊號(D)的正確性，參考圖 9 GPSDC 電路輸出比對訊號時序規格。

2.3.1 半正矢公式 (Haversine Formula)

半正矢公式是一種根據兩點的經度和緯度來確定大圓上兩點之間距離的計算方法，在導航有著重要地位。它是球面三角學中「半正矢定理」公式的特例，該定理涉及了球面三角形的邊和角，如圖 1 所示。其中 φ 表示 latitude(緯度)， λ 表示 longitude(經度)， R 為地球東西直徑為 12756274km，在運算過程中須將度 (degree) 轉換成弧度 (radian) 運算，文後 radian 以 rad 表示。1 度 (degree) = 0.017453292 弧度 (rad)，其中 rad 整數使用 0 bit，小數使用 16bit 表示為 16'h477。

定義說明：

A：第一點座標位置

B：第二點座標位置

φA ：第一點座標位置緯度

φB ：第二點座標位置緯度

λA ：第一點座標位置經度

λB ：第二點座標位置經度

半正矢定理公式： $\text{haversin}(\theta) = \sin^2\left(\frac{\theta}{2}\right)$ 。

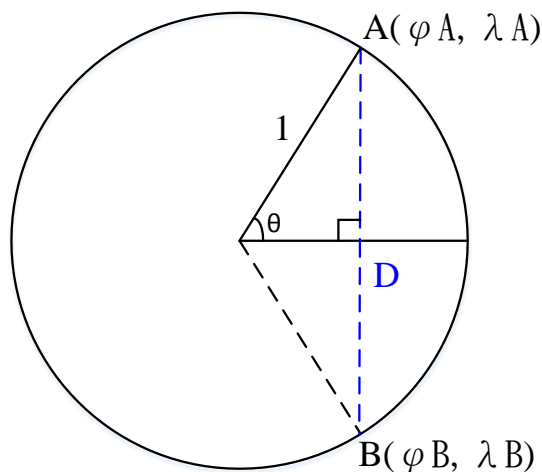


圖 3. 半正矢定理

公式 1：半正矢定理運算式

運算說明：

- 已知 $\text{rad}=16'h477$ (0.017453292)、 $R=12756274$ 。
- 將 rad 及 φA 、 φB 、 λA 、 λB 代入中繼運算式，可得 a 中繼運算資料。
- 將中繼運算結果 a，代入距離運算式中，可得到兩點座標距離資料。
- 其中，進行 $\cos(x)$ 及 $\sin^{-1}(\sqrt{a})$ 運算時，必須使用線性內插公式運算後之數值(參考 2.3.5 公式 2)。

$\text{rad}=16'h477$ ， $R=12756274$ 。

中繼運算式： $a = \sin^2\left(\frac{(\varphi B - \varphi A) * \text{rad}}{2}\right) + \cos\varphi A * \cos\varphi B * \sin^2\left(\frac{(\lambda B - \lambda A) * \text{rad}}{2}\right)$

距離運算式： $D=R*\sin^{-1}(\sqrt{a})$

註：中繼運算請參考 2.3.6 說明

2.3.2 sine 函數值

在本題電路中，sine 函數因 rad 值太小，故 sine 函數運算結果近似等於 sine 函數輸入值即 $\sin(a)=a$ 。

範例： $\sin(0.0003041811287403106689453125) = 0.0003041811287403106689453125$ 。

2.3.3 cosine 值查表方式

運算半正矢公式的過程中，將會需要用到 cosine 的運算值，在 testbench 中已建立好 cosine 函數定值的運算結果並儲存在 cos_rom 模組裡，故不允許使用 DesignWare® cosine 函數 IP，cosine 表格數值為遞增表示。

Cosine rom 訊號線有位址線及資料線，COS_ADDR 為 7 位元寬度位址線，COS_DATA 為 96 位元寬度資料線，如圖 4 所示，其中 COS_DATA[95:48] 為 cosine 函數的輸入值，其整數位皆為 16 位元，小數位為 32 位元，COS_DATA[47:0] 為 cosine 函數輸出值，其整數位皆為 16 位元，小數位為 32 位元表示。當位址線為 0 時，表示 cosine 函數輸入值 48'h18C7AE147A (2.478000e+001)，cosine 函數輸出值為 48'hF03CE5C7 (9.384292e-001)，即 $\cos(2.478000e+001) = 9.384292e-001$ ，以此類推，其內容如表二。

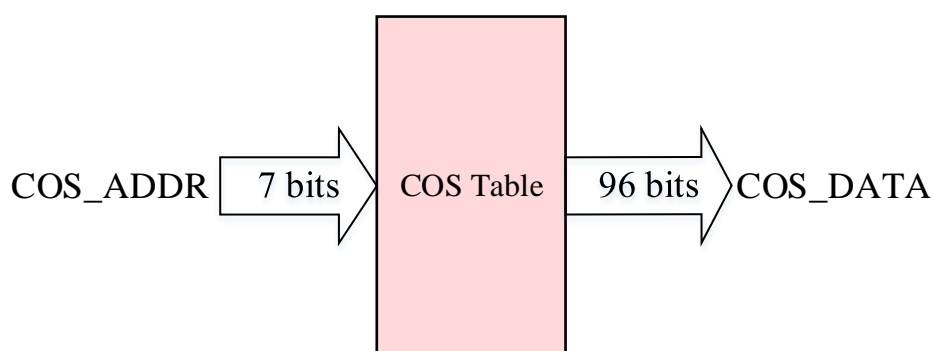


圖 4. cosine rom 方塊圖

表二、cosine 函數表

address	x	cos(x)
COS_ADDR	COS_DATA[95:48]	COS_DATA[47:0]
0	48'h18C7AE147A	48'hF03CE5C7
1	48'h18C7B47304	48'hF03F1902
2	48'h18C7BAD18D	48'hF0414C17
.	.	.
.	.	.
.	.	.
125	48'h18CACA3D70	48'hF14B6738
126	48'h18CAD09BF9	48'hF14D87D2
127	48'h18CAD6FA82	48'hF14FA845

2.3.4 $\sin^{-1}(\sqrt{x})$ 值查表方式

Testbench 中已建立好 $\sin^{-1}(\sqrt{x})$ 函數定值的運算結果，並儲存在 asin_rom 模組裡，故不允許使用 DesignWare® arcsin 函數 IP，arcsin 表格數值為遞增表示。

asin_rom 訊號線有位址線及資料線，ASIN_ADDR 為 6 位元寬度位址線，ASIN_DATA 為 128 位元寬度資料線，如圖 5 所示，其中 ASIN_DATA[127:64] 為 $\sin^{-1}(\sqrt{x})$ 函數的輸入值，其整數位為 0 位元，小數位為 64 位元，ASIN_DATA[63:0] 為 $\sin^{-1}(\sqrt{x})$ 函數輸出值，其整數位為 0 位元，小數位為 64 位元表示，當位址線為 1 時，表示 $\sin^{-1}(\sqrt{x})$ 函數輸入值 64'h000000855305 (4.736630e-13)， $\sin^{-1}(\sqrt{x})$ 函數輸出值為 64'h0b8beee546d2 (6.882318e-07)，即 $\sin^{-1}(\sqrt{4.736630e-13})=6.882318e-07$ ，以此類推，其內容如表三。

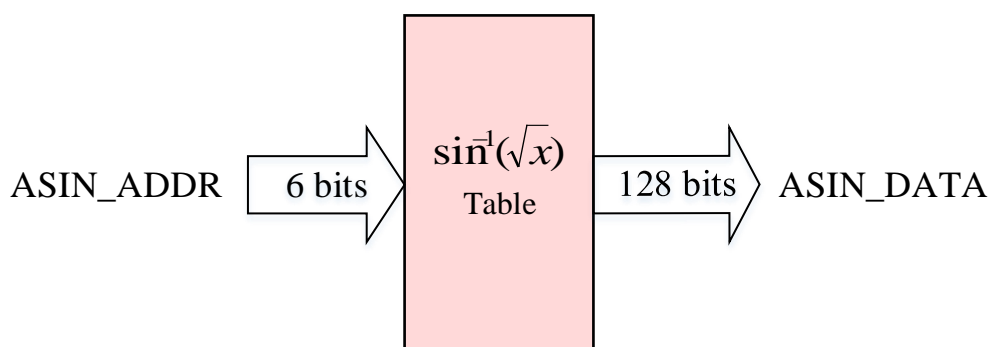


圖 5. asin_rom 方塊圖

表三、 $\sin^{-1}(\sqrt{x})$ 函數表

address	x	$\sin^{-1}(\sqrt{x})$
ASIN_ADDR	ASIN_DATA[127:64]	ASIN_DATA[63:0]
0	64'h0	64'h0
1	64'h855305	64'hB8BEEE546D2
2	64'h10AA60A	64'h105452200523
3	64'h18FF90F	64'h13FFD39301E1
.	.	.
.	.	.
.	.	.
58	64'h1E34CF22	64'h57EFB769F70E
59	64'h1EBA2227	64'h58B0F4209C7C
60	64'h1F3F752C	64'h59708F5C67BE
61	64'h1FC4C831	64'h5A2E93825D1C
62	64'h204A1B36	64'h5AEB0A8AE0A4
63	64'h20CF6E3B	64'h5BA5FE07D1ED

2.3.5 線性內插方式

$\cos(x)$ 及 $\sin^{-1}(\sqrt{x})$ 表皆為定值輸入、輸出，在運算過程中的結果，無法等於 $\cos(x)$ 及 $\sin^{-1}(\sqrt{x})$ 的輸入值，故參賽者必須利用兩點線性內插公式，如公式 2 所示，算出對應的數值，其中 cosine 函數使用整數位為 16 位元寬度，小數位為 32 位元寬度表示進行查表； $\sin^{-1}(\sqrt{x})$ 運算內插請使用整數位為 0 位元寬度，小數位為 64 位元寬度表示進行查表，如圖 7 所示。

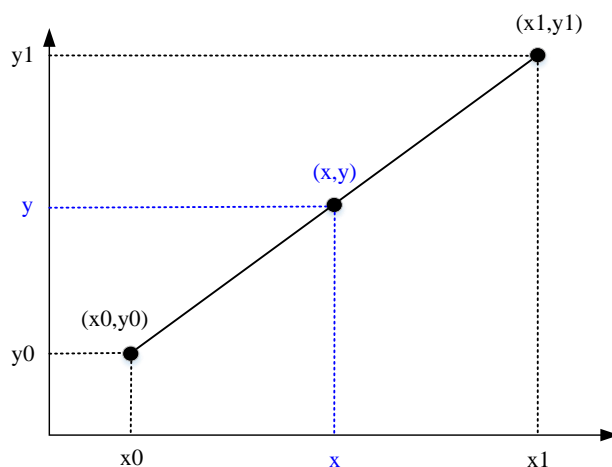


圖 6. 線性內插

公式 2：線性內插值公式

$$\frac{y-y_0}{x-x_0} = \frac{y_1-y_0}{x_1-x_0}$$

即移項可得

$$y = y_0 + (x - x_0) * \frac{y_1 - y_0}{x_1 - x_0} \quad \dots\dots\dots \text{①式}$$

$$y = \frac{y_0 * (x_1 - x_0) + (x - x_0) * (y_1 - y_0)}{(x_1 - x_0)} \quad \dots\dots\dots \text{②式}$$

註：①式與②式為相同算式，但為避免 truncation 所造成誤差，故必須使用②式進行運算

■cos(x)範例說明：

- 若欲求 $\cos(24.78459167_{(10)})$ 之內插值
- 已知 $24.78459167_{(10)} = 48'h18c8db0000$
- 透過查表可知， $48'h18c8db0000$ 介於 $48'h18C8D96FA8$ 與 $48'h18C8DFCE31$ 之間
- 求出 $\cos(24.78459167_{(10)})$ 之內插值 $96'hf0a435481bf84f2c$ 後，再帶入公式 1 中。

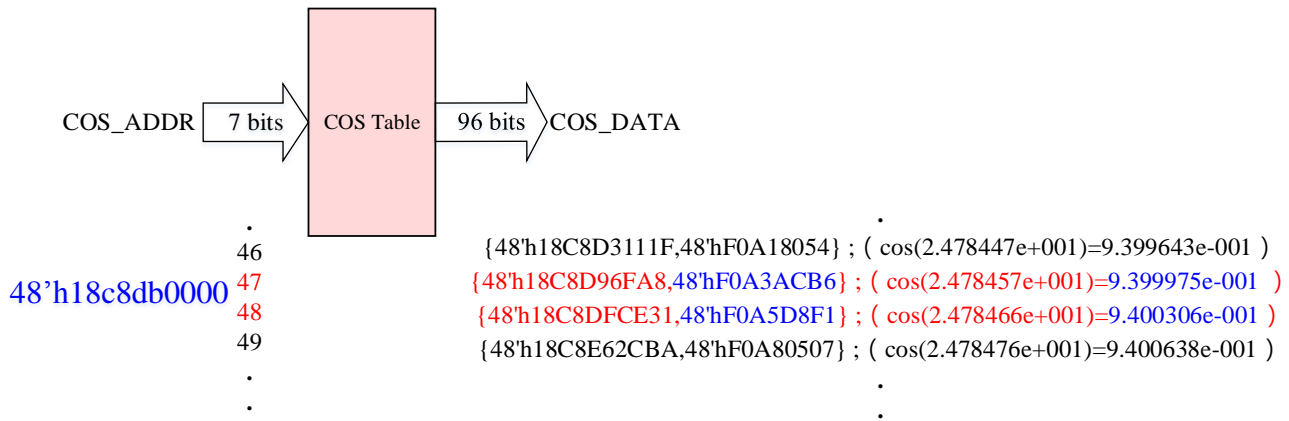


圖 7. 線性內插表查

運算說明如下：

令

$x = 48'h18c8db0000$

$(x_0, y_0) = (48'h18C8D96FA8, 48'hF0A3ACB6)$

$(x_1, y_1) = (48'h18C8DFCE31, 48'hF0A5D8F1)$

將 x 、 x_0 、 y_0 、 x_1 、 y_1

代入公式 2 中，求出內插 y 值 $96'hf0a435481bf84f2c$ （其中整數位 32 位元，小數為 64 位元表示）
即 $\cos(24.78459167_{(10)})$ 的內插值為 $96'hf0a435481bf84f2c$ （ $0.94000561_{(10)}$ ）。

2.3.6 中繼運算值明說

$$\text{中繼運算式：} a = \sin^2 \left(\frac{(\varphi_B - \varphi_A) * \text{rad}}{2} \right) + \cos \varphi_A * \cos \varphi_B * \sin^2 \left(\frac{(\lambda_B - \lambda_A) * \text{rad}}{2} \right)$$

其中，令

$$\frac{(\varphi_B - \varphi_A) * \text{rad}}{2} \dots\dots\dots \text{①式}$$

$$\frac{(\lambda_B - \lambda_A) * \text{rad}}{2} \dots\dots\dots \text{②式}$$

必須先進行，①、②式的運算，再進行 $\sin^2(x)$ 運算

2.4 系統時序規格圖

本電路的 clock 訊號為 **clk 正緣觸發** 工作，reset_n 訊號為低位準啟動，reset_n 訊號持續 3 個時脈週期後結束，系統時序規格如圖 8GPSDC 電路初始化輸入之時序，圖 9GPSDC 電路資料比對之時序圖所示。

1. T0~T1 時間點，GPSDC 電路初始化。
2. T2 時間點，當 clk 正緣時，DEN 為 high，表示目前的 LAT_IN、LON_IN 為有效資料，每筆有效資料將持續 1 個 clock cycle，GPSDC 電路須於 clk 正緣時抓取有效資料。
3. T3~T4 將間隔 256 個 cycle，系統將輸入第 2 點座標資料。

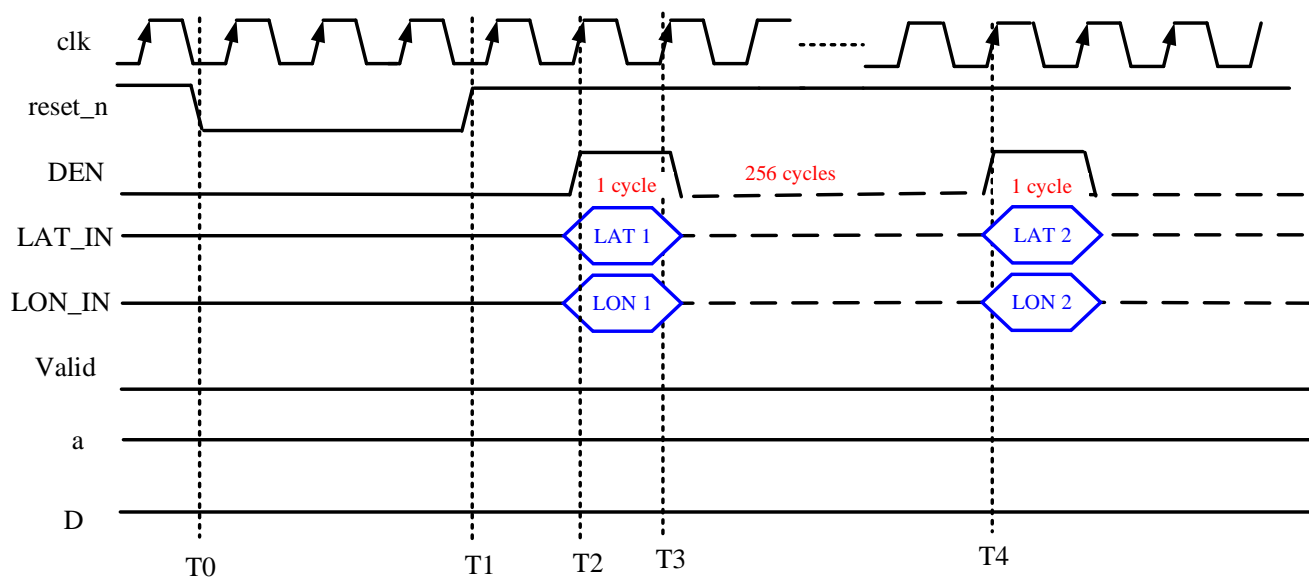


圖 8. GPSDC 電路初始化輸入之時序圖

4. T5 時間點，表示 GPSCD 兩點座標運算完畢時，將 Valid 拉為 high，並將 Valid、a、D 訊號維持 1 個 clock cycle。
5. T6 時間點，當 testbench 偵測到 Valid 為 high 時，系統將同時進行 a、D 訊號的比對。
6. T7 時間點，當 testbench 於 T6 時間點偵測到 Valid 為 high，經過 1 個 clock cycle 於 T7 時間點輸出第 3 筆座標資料。
7. T8~T10 時間點，重覆 T5~T7 步驟，輸出第 4 點座標資料，第 5 點座標資料以此類推。

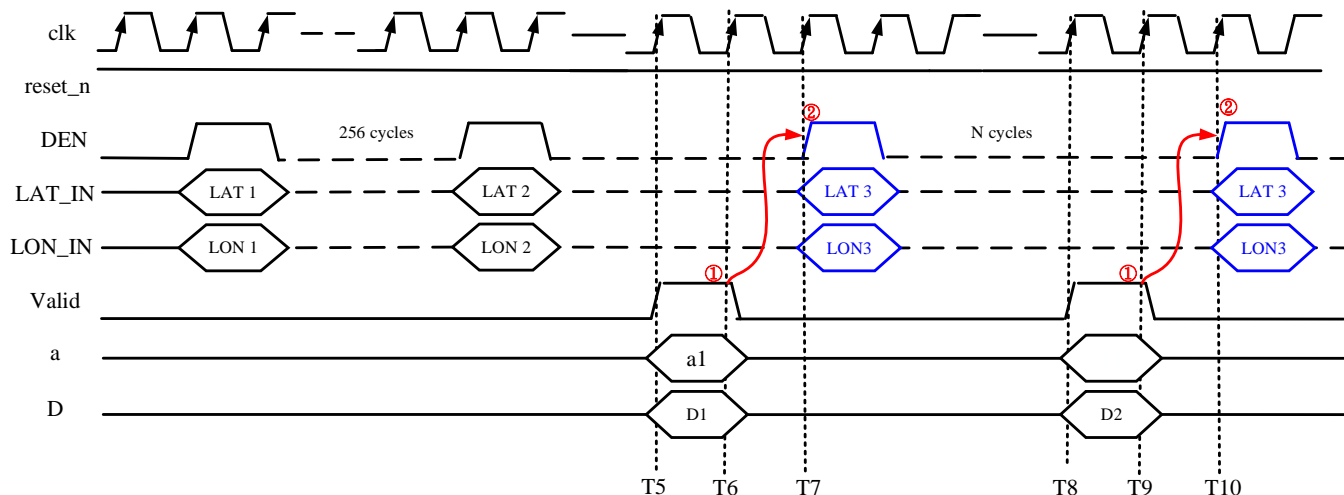


圖 9. GPSCD 電路輸出比對訊號

2.4.1 COS、ASIN ROM 時序規格

本系統之 testbench 查表記憶體，故當位址線切換時，資料輸出無延遲時間。

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，**不允許使用 DesignWare® IP 進行設計**。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，並完全符合主辦單位所提供的標準設計結果。

✧ 評分項目一：依”模擬時間”(Time)長短評分

各參賽隊伍執行 RTL Simulation 及 Pre-layout Gate-level Simulation 模擬完後，會出現模擬時間，評分人員會以此模擬時間如下面範例，紀錄成 **Time = 987210 NS** 做評分。

```
-----
TEST START !!!
-----
*Verdi3* FSDB: For performance reasons, the Memory Size Limit has been increased
to 128M.
All data have been generated successfully!
-----PASS-----
Simulation complete via $finish(1) at time 987210 NS + 0
./testbench.v:145      #10 $finish;
```

✧ 評分項目二：依”面積”(Area)大小評分

各參賽隊伍完成電路合成後，Cell Area 可利用以下指令產生而得知

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

以 RC 產生 QoR report 的指令： report qor > GPSDC.qor

評分人員會以此 GPSDC.qor 如下面範例，紀錄成 **Area = 218075** (以小數點後做四捨五入)做評分。

```
Area
-----
Combinational Area:      66483.764989
Noncombinational Area:
                          151591.403398
Buf/Inv Area:            9919.605447
Total Buffer Area:       9715.92
Total Inverter Area:     203.69
Macro/Black Box Area:    0.000000
Net Area:                1834056.160553
-----
Cell Area:               218075.168386
Design Area:            2052131.328939
```

✧ 評分方法及標準

評分方式會依設計完成程度，分成 A、B、C 三種等級，排名順序為 **A > B > C**。主辦單位將根據設計內容的完成度給予記分。

1. **A 等級**：RTL 與 Gate-level simulation 結果完全正確。

此等級之成績計算方式如下：

$$\text{Score} = \text{Time} \times \text{Area}$$

註: Time 以 Gate-level simulation 時間計算。

註:本等級中，Score 越小者為同級名次越好!

2. **B 等級**：RTL simulation 結果完全正確，但 Gate-level simulation 結果出現錯誤。

在 clock cycle 設定為 10ns 的情況下，此等級之成績計算方式如下：

$$\text{Score} = \text{Time}$$

註:本等級中，Score 越小者為同級名次越好!

3. **C 等級**：RTL simulation Valid、a、D 訊號 結果部分正確。

在 clock cycle 設定為 10ns 的情況下，此等級之成績計算方式如下：

$$\text{Score} = \text{a 訊號錯誤數}$$

註:本等級中，Score 越小者為同級名次越好!

附錄

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，亦即參賽者必須繳交的檔案資料；

附錄 D 則為設計檔整理步驟說明；

附錄 E 則為設計資料庫說明。

附錄 A 軟體環境

競賽所提供的設計軟體環境如下表四。驗證評分時，係以所列軟體作為驗證依據。

表四、設計軟體環境

Vendor	Tool	Executable
Cadence	Virtuoso *1	icfb
	Composer	icfb
	NC-Verilog	ncverilog
	SOC Encounter	encounter
Synopsys	Design Compiler	dv , dc_shell
	VCS-MX	vcs
	IC Compiler	icc_shell -gui
	Hspice	hspice
	Cosmos Scope *1	cscope
	Custom Explorer *1	wv
	Laker *1	laker
	Laker ADP*1	adp
	Verdi *1	verdi , nWave , nLint
Mentor	Calibre *3	calibre
	ModelSim	vsim
Utility	vi	vi , vim
	gedit	gedit
	nedit	nedit
	pdf reader	acroread
	calculate	gnome-calculator , bc -l
	gcc	gcc

EDA 軟體所需使用的 license 皆已設定完成，不須額外設定

*1 表示該軟體限定使用 1 套 license

*3 表示該軟體限定使用 3 套 license

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
testbench.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及預期輸出信號。
GPSDC.v	參賽者所使用的設計檔，已包含系統輸/出入埠之宣告
lat.dat、lon.dat	測試樣本檔案
a_Pattern.dat	中繼運算結果比對樣本檔案
D_Pattern.dat	距離資料比對樣本檔案
GPSDC.sdc	Design Compiler 電路合成規範檔。 本規範檔除了 cycle 可修改外，其餘 constraints 皆不可修改。
tsmc13_neg.v	Gate-level simulation 所需要之 cell library file
synopsys_dc.setup	Design Compiler 初始設定範例檔案

2. 本題所提供的 Testbench 檔案，有多增加幾行特別用途的敘述如下：

```
`define sdf_file    “./GPSDC_syn.sdf”

`ifdef SDF
    initial $sdf_annotate(`sdf_file , u_GPSDC);
`endif
```

註：

1. SDF 檔案，請自行修改 SDF 實際檔案名稱及路徑後再模擬。
2. 在 Testbench 中，本題提供 `ifdef SDF 的描述，目的是讓 Testbench 可做為 RTL 模擬及合成後模擬皆可使用。當參賽者在合成後模擬時，請務必多加上一個參數 +define+SDF，方可順利模擬。
3. 合成後的 Gate-Level Simulation 模擬指令範例如下：

```
ncverilog testbench.v GPSDC_syn.v -v tsmc13_neg.v +define+SDF
```

3. 比賽共提供一組測試樣本，參賽者可依下面範例來進行模擬：

主辦單位於評分時，將可能額外再使用其他組測試樣本進行評分工作。

RTL Simulation 時使用指令如下：

- 使用 ncverilog 模擬指令範例如下：

```
ncverilog testbench.v GPSDC.v
```

- 使用 modelsim 模擬，則是在 compiler verilog 時，使用下面指令：

```
vlog testbench.v
```

Gate-Level Simulation 時使用指令如下：

- 使用 ncverilog 模擬指令範例如下：

```
ncverilog testbench.v GPSDC_syn.v -v tsmc13_neg.v +define+SDF
```

define 中加上 SDF 可讓測試程式引入 gate level netlist 的 sdf 檔案資訊。

- **NC-Verilog 使用者若想要輸出 FSDB 格式波形檔案，**

可自行再加入 +define+FSDB +access+r

modelsim 使用者，請直接使用內建波形來進行除錯。

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，請務必將使用到的各 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。

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.v	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

附錄 D 檔案整理步驟

當所有文件準備齊全如表四所列，請按照以下步驟指令，提交相關設計檔案，將所有檔案複製到同一資料夾下，步驟如下：

1. 在自己的 home directory 建立一個新目錄，名稱叫做“result”，範例如下：

```
> mkdir ~/result
```

2. 將附錄 C 中要求的檔案全部複製到 result 這個目錄。範例如下：

```
> cp GPSDC.v ~/result/
```

```
> cp GPSDC_syn.v ~/result/
```

其他檔案依此類推.....

3. 在 Design Report Form 中，填入所需的相關資訊。

附錄 E 設計資料庫

設計資料庫位置： /usr/cad/icc2019/CBDK_IC_Constest_v2.1

目錄架構

SynopsysDC/
db/

slow.db

Synthesis model (slow)

lib/

slow.lib

timing and power model

Verilog/

tsmc13_neg.v

Verilog simulation model

Design Report Form

登入帳號(login-id)		
RTL category		
<i>Design Stage</i>	<i>Description</i>	<i>File Name</i>
RTL Simulation	使用之 HDL 名稱 (例如：Verilog、VHDL)	
RTL Simulation	RTL 檔案名稱 (RTL Netlist file name)	
Gate-Level category		
<i>Design Stage</i>	<i>Description</i>	<i>File Name</i>
Pre-layout Gate-level Simulation	Gate-Level 檔案名稱 (Gate-Level Netlist file name) (*_syn.v)	
	Pre-layout sdf 檔案名稱 (*_syn.sdf)	
	Gate-Level simulation, 所使用的 CYCLE Time (請確定模擬功能正確)	() ns
	Cell Area (以小數點後做四捨五入)	
Over All	最後完成之等級? (ex: 等級 A)	
	若為 C 等級, 請填入錯誤數	
其他說明事項 (Any other information you want to specify: (如設計特點 ...)) 如寫不下可寫於背面		