

初賽時間為 3 月 20 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/IcContest2013ByNcu>)，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](mailto: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

# 2013 IC Design Contest Preliminary

## 標準元件數位電路設計 - 研究所組

### *Frequency Analysis System*

#### 1. 問題描述

近來加速度感測器被廣泛地被應用在 3C 產品、通訊設備、車輛安全、醫療電子、工業監控…等方面，其中以加速度感測器的應用更為常見；加速度感測器可提供加速度、速度及位移的資訊，可以進行時域（time domain）、頻域（frequency domain）資料的分析，依據其提供的資料可分析、檢測出機械設備的軸承的平衡、故障…等問題，用以預防機械設備的嚴重損壞及提高產品的可靠度及安全性。

本題目運算分析電路，是一個應用在感測系統中，判斷儀器、設備運作震動訊號的運算分析電路。系統前端將會讀取感測電路所傳出的重力加速度值，由此重力加速度值所產生的時域訊號，經過濾波器（Filter），鎖定儀器、設備的震動頻率範圍，並將環境所產生的雜訊及高頻訊號過濾，並由 FFT（快速傅利葉轉換）電路，將時域訊號轉換為頻域訊號，最後分析頻率資訊，找出主要頻段，[根據此頻段資訊，可進一步做特徵分析來判別此系統是否震動太大須暫時停機維修\(應用在工業儀器運作監控\)或是緊急煞車所造成之加速度\(應用在車輛安全\)等許多應用。](#)

請完成一有限脈衝響應濾波器（Finite Impulse Response Filter，FIR Filter）、一快速傅利葉轉換（FFT：Fast Fourier transform）及一頻譜分析電路（Analysis）。

本系統有 4 只信號輸入（clk, rst, data, data\_valid）、21 只信號輸出（fir\_d, fir\_valid, fft\_d0~fft\_d15, fft\_valid, done, freq），如圖一所示，關於各輸入/輸出信號的功能說明，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。

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，測試檔不予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

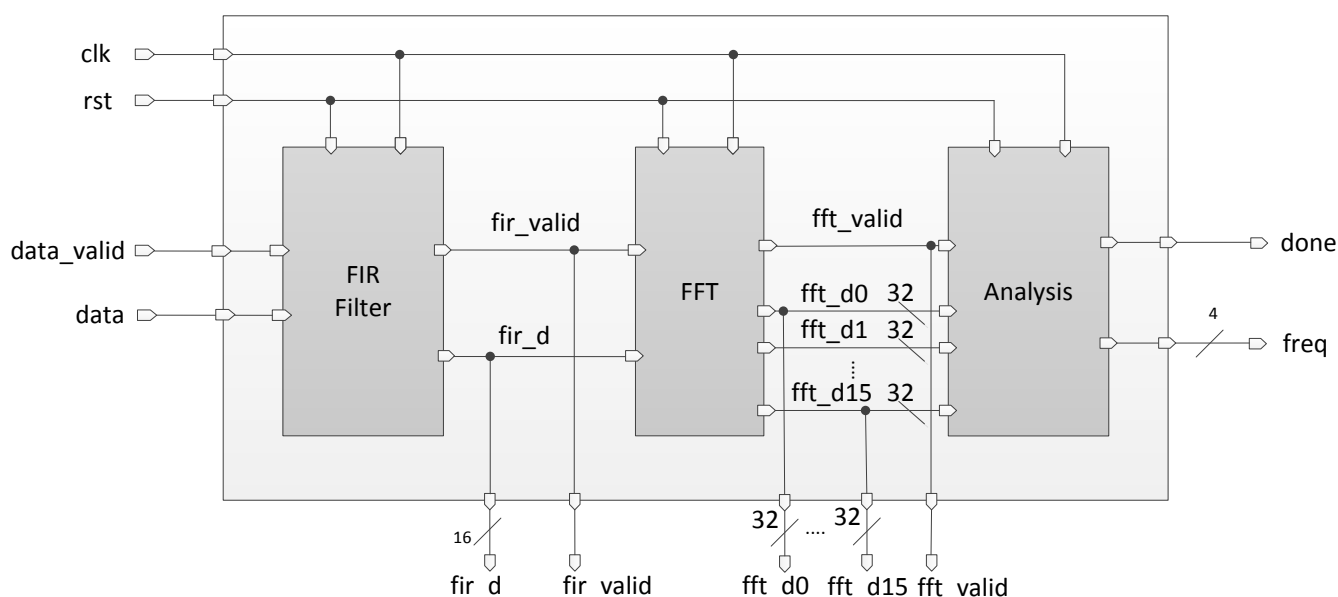


圖 一、系統方塊圖

### 2.2 輸入/輸出介面

表 一、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	時脈信號。 說明：主控端與內部記憶體於時脈正緣時動作。
<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>data_valid</i>	input	1	資料預備信號。 說明：當主控端預備傳送資料時，會將此信號設為 high。
<i>data</i>	input	16	主控端時域訊號輸入。
<i>done</i>	output	1	完成指令輸出信號。 說明：當電路完成系統運算時，將 done 設為 high 表示完成。

<i>fir_d</i>	output	16	16 位元 FIR filter 之資料輸出訊號。
<i>fir_valid</i>	output	1	FIR filter 資料有效信號。
<i>fft_d0 ~ fft_d15</i>	output	32	32 位元 FFT 之資料輸出訊號。 說明：16 個 32 位元 FFT 之輸出訊號。
<i>fft_valid</i>	output	1	FFT 資料有效信號。
<i>freq</i>	output	4	4 位元主要頻段輸出訊號。

## 2.3 系統功能描述

本系統從主控端傳送時域資料(參考圖六)，經過有限脈衝響應濾波器(Finite Impulse Response Filter, FIR Filter)，將高頻訊號及雜訊濾除掉，並進行快速傅利葉轉換(FFT: Fast Fourier transform)，將時域訊號轉為頻域訊號，最後自行設計一頻譜分析電路，找出此段訊號的主要頻段為何。

如圖二範例，主控端的訊號為 17Hz sin 訊號混合 200Hz sin 訊號，如圖二所示，

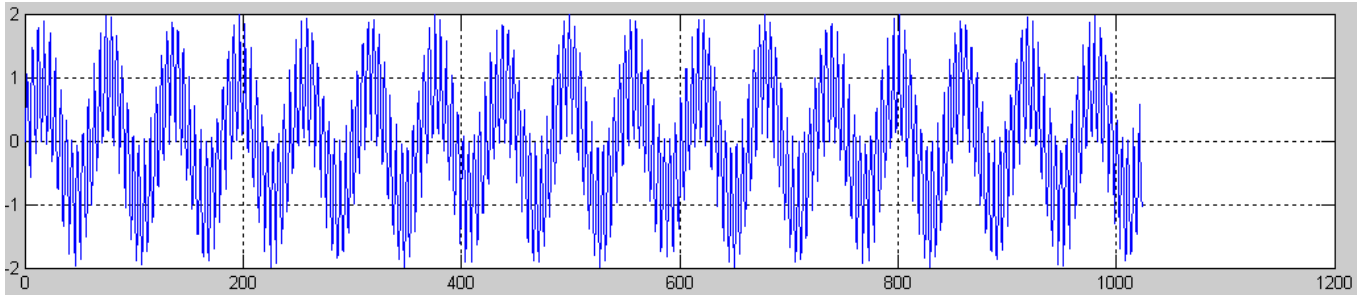


圖 二、主控端時序訊號

經過 FIR 濾波器後，其 200Hz Sin 訊號將被濾除、削減，如圖三所示。

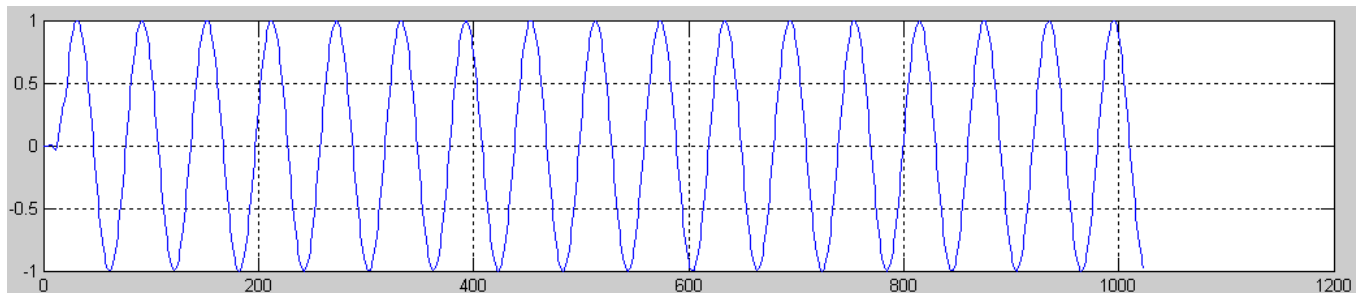


圖 三、FIR 濾波器後之訊號

在傳送濾波後的訊號時，將 *fir\_vaild* 信號拉為 high，*fir\_d* 使資料能夠一個時脈週期傳送一筆資料(參考圖七)，經過 FFT 處理後會得到頻域訊號，如圖四所示。將 *fft\_valid* 信號拉為 high，*fft\_d* 使資料能夠一個時脈週期傳送一筆資料至頻譜分析電路(參考圖八)，完成分析後，將 *done* 信號拉為 high，*freq* 輸出主要頻段為第 0 頻段 (*freq=4'b0000*) (參考圖四)。

FFT 共有 *fft\_d0*~*ddt\_d15* 共 16 個平行輸出訊號，其中 *fft\_d0* 代表的即為第 0 頻段 (*freq=4'b0000*)、*fft\_d1* 代表第 1 頻段 (*freq=4'b0001*)、而 *fft\_d15* 代表第 15 頻段 (*freq=4'b1111*)，其他依此類推。

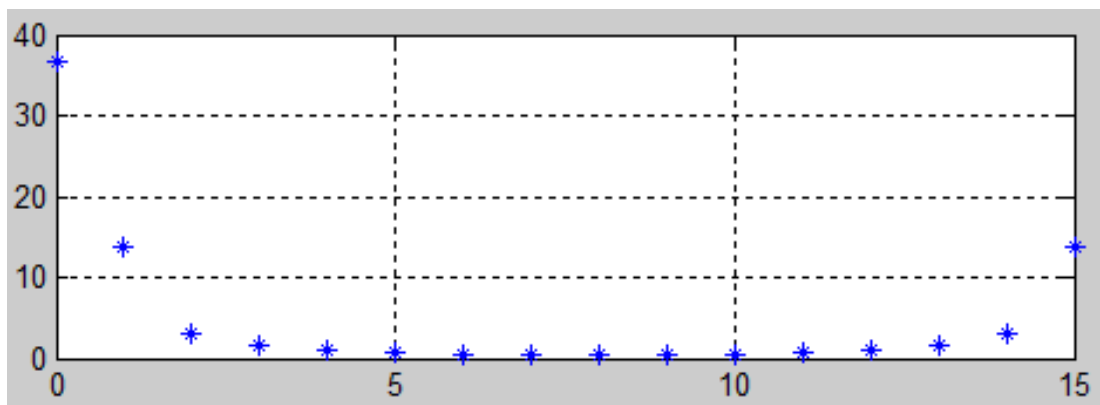


圖 四、FFT 處理後之頻譜

## 2.4 時序規格圖

本系統時序規格，共有系統時脈、重置及完成訊號時序規格、主控端資料傳輸時序規格、FIR 濾波器輸出有效資料比對時序規格、FFT 輸出有效資料比對時序規格，四個部分。

### 2.4.1 系統時脈、重置及完成訊號時序規格

主控端時脈與重置訊號的時序波形如圖五所示，其中的時脈週期  $t_{\text{CYCLE}}$  預設為 **20ns**。當系統啟動時，**串列**輸入訊號首先將先經過 FIR 處理後由 fir\_d 及 fir\_valid **串列**輸出。FIR 訊號輸出同時，將此輸出訊號依每 16 筆為一組轉為**並列**訊號，接著將每組**並列**訊號進行 FFT 訊號處理，FFT 處理後之訊號使用 fft\_d0~fft\_f15 及 fft\_valid **並列**輸出。接著將每 16 點 FFT 資料處理並進行頻率分析後，將結果輸出 (*freq*)，並將 *done* 信號設定為 high 表示此組訊號處理完成。其中，fir\_valid、fft\_valid 及 done 等訊號皆為輸出有效提示訊號，週期皆為 1 clock cycle。另外，FIR 有效輸出的**串列**輸出訊號轉成 FFT **並列**輸入訊號時，訊號間將不做任何 overlap，例如：第 1 筆 FFT **並列**輸入訊號組成成份為 fir\_d(0)~fir\_d(15)，第 2 筆 FFT **並列**輸入訊號組成成份為 fir\_d(16)~fir\_d(31)，其他依此類推。以上說明可參考圖 5。本系統將由主控端輸入 1024 筆資料，故會有 64 次運算結果。

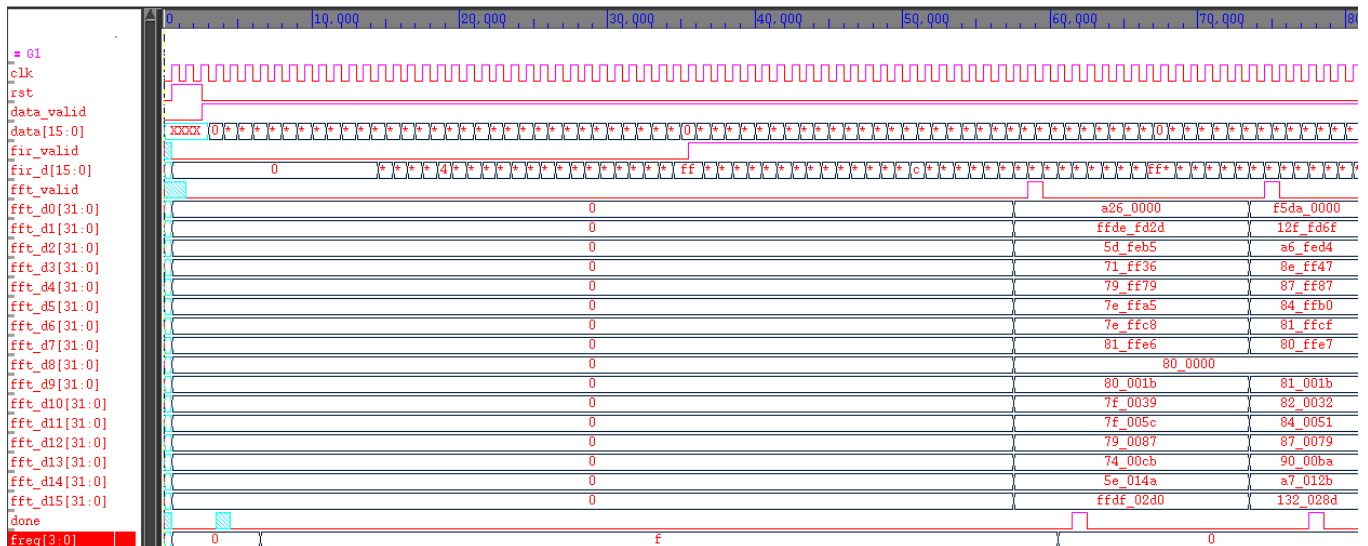


圖 五、系統時序圖

## 2.4.2 主控端資料傳輸時序規格

當主控端將 *data\_valid* 訊號設定為 high 時，主控端 *data* 埠，將會在每一個時脈送出一筆時域資料，其時序規格如圖六所示。主控端 *data* 訊號之資料格式為 16bits，含 1bit 的 sign bit、7 bits 的整數資料及 8bits 的小數資料，組成方式可參考圖十所示。

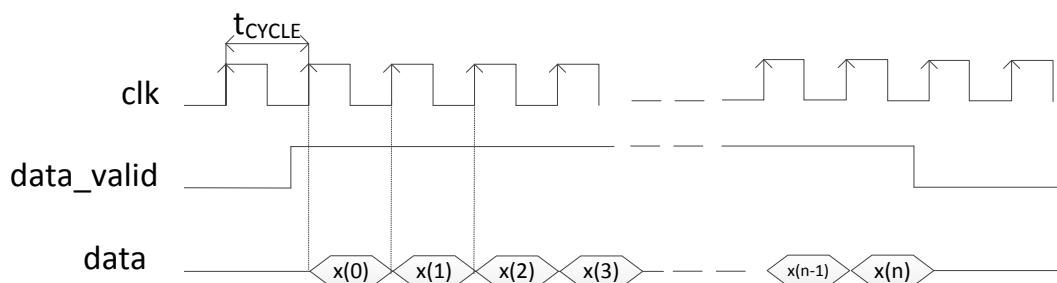


圖 六、主控端資料傳輸時序圖

## 2.4.3 FIR 濾波器輸出有效資料比對時序規格

當資料經過 FIR 濾波器處理後，會將處理後的資料輸出至 FFT 電路，在資料傳輸的過程中，若 *fir\_valid* 信號設為 high，則表示 *fir\_d* 始傳送資料至 FFT 電路，測試程式將同時進行資料比對，資料比對時序如圖七所示。

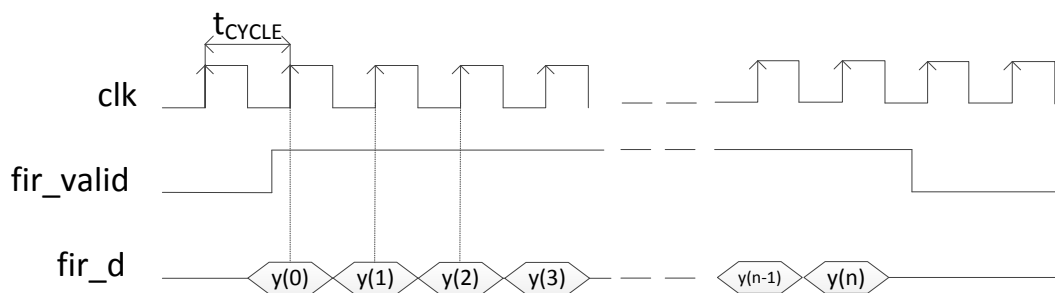


圖 七、FIR 端資料比對時序圖

## 2.4.4 FFT 輸出有效資料比對時序規格

當資料經過 FFT 處理後，會將處理後的資料輸出至頻譜分析電路，在資料傳輸的過程中，若 `fft_valid` 信號設為 high，則表示 `fft_d0~fft_d15` 開始傳送資料至頻譜分析電路，測試程式將同時進行資料比對，其資料比對時序如圖八所示。

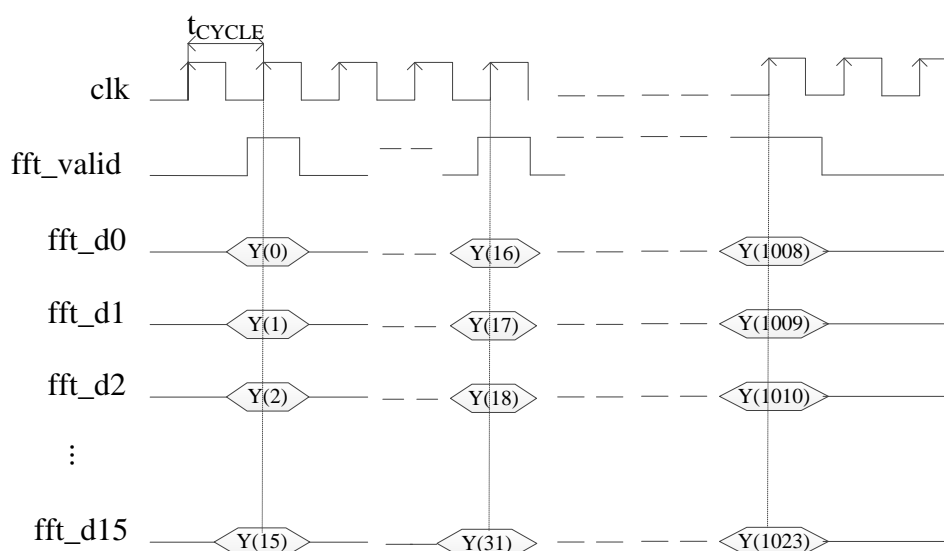


圖 八、FFT 端資料比對時序圖



## 2.5 有限脈衝響應濾波器（Finite Impulse Response Filter，FIR Filter）功能描述

本系統中所使用的有限脈衝響應濾波器（Finite Impulse Response Filter，FIR Filter）為 **31 階低通濾波器（32 個係數）**，此濾波器主要功能為濾除不必要的高頻訊號，其濾波器係數為固定之係數，其係數如表二所示。**須注意，有效輸出資料為主控端第 32 筆進入 FIR filter 電路開始計算。**

單一頻道的 FIR 濾波器可用公式一表示，硬體架構如圖九所示，其 FIR 資料格式如圖十所示。

Low-pass Filter Coefficient ( h )			
h(0)	-0.001505748051548	h(16)	0.229154203266836
h(1)	-0.001868548463782	h(17)	0.186019113110601
h(2)	-0.001366448872269	h(18)	0.115911664195512
h(3)	9.086560980884849e-04	h(19)	0.043365841112764
h(4)	0.005060978234550	h(20)	-0.009960448073993
h(5)	0.008948776436908	h(21)	-0.033966171510252
h(6)	0.008342764987706	h(22)	-0.032084609166509
h(7)	-4.333516292017973e-04	h(23)	-0.016526671675409
h(8)	-0.016526671675409	h(24)	-4.333516292017973e-04
h(9)	-0.032084609166509	h(25)	0.008342764987706
h(10)	-0.033966171510252	h(26)	0.008948776436908
h(11)	-0.009960448073993	h(27)	0.005060978234550
h(12)	0.043365841112764	h(28)	9.086560980884849e-04
h(13)	0.115911664195512	h(29)	-0.001366448872269
h(14)	0.186019113110601	h(30)	-0.001868548463782
h(15)	0.229154203266836	h(31)	-0.001505748051548

表 2. LPF 係數（表 2 之內容已存放於 FIR\_coefficient.dat）

$$y(n) = \sum_{k=0}^{N-1} h(k) x(n-k)$$

公式一

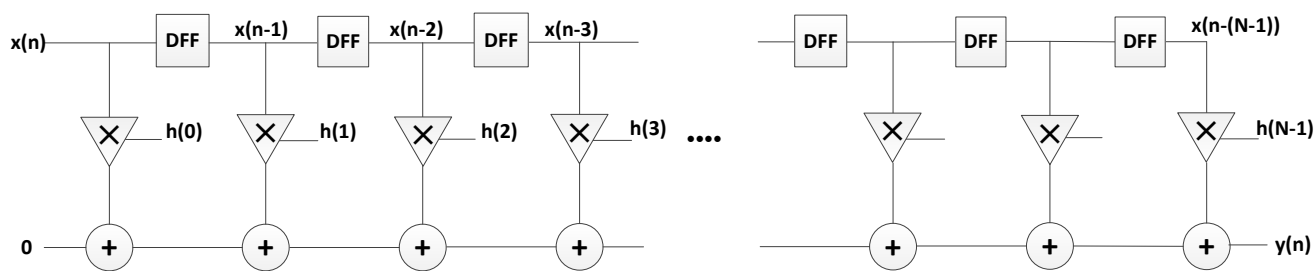


圖 九、FIR 濾波器硬體架構

正/負	整數	小數
1bit	7bit	8 bit

圖 十、FIR 資料格式 (data, fir\_d)

## 2.6 快速傅利葉轉換 (FFT：Fast Fourier transform)功能描述

本題目中所使用的快速傅利葉轉換 (FFT：Fast Fourier transform)，本系統規格須完成 16 點的快速傅利葉轉換，硬體架構如圖十一所示。此快速傅利葉轉換電路，是為將時域訊號轉換為頻域訊號，以利後續之訊號分析及處理，其原始數學表示式如公式二所示。(公式推導參考附錄 C)

$$Y(k) = \sum_{m=0}^{N-1} w^{mk} y(m) \quad k = 0, 1, \dots, N-1$$

$$w = e^{-2j/N}$$

$$j = \sqrt{-1}$$

公式二

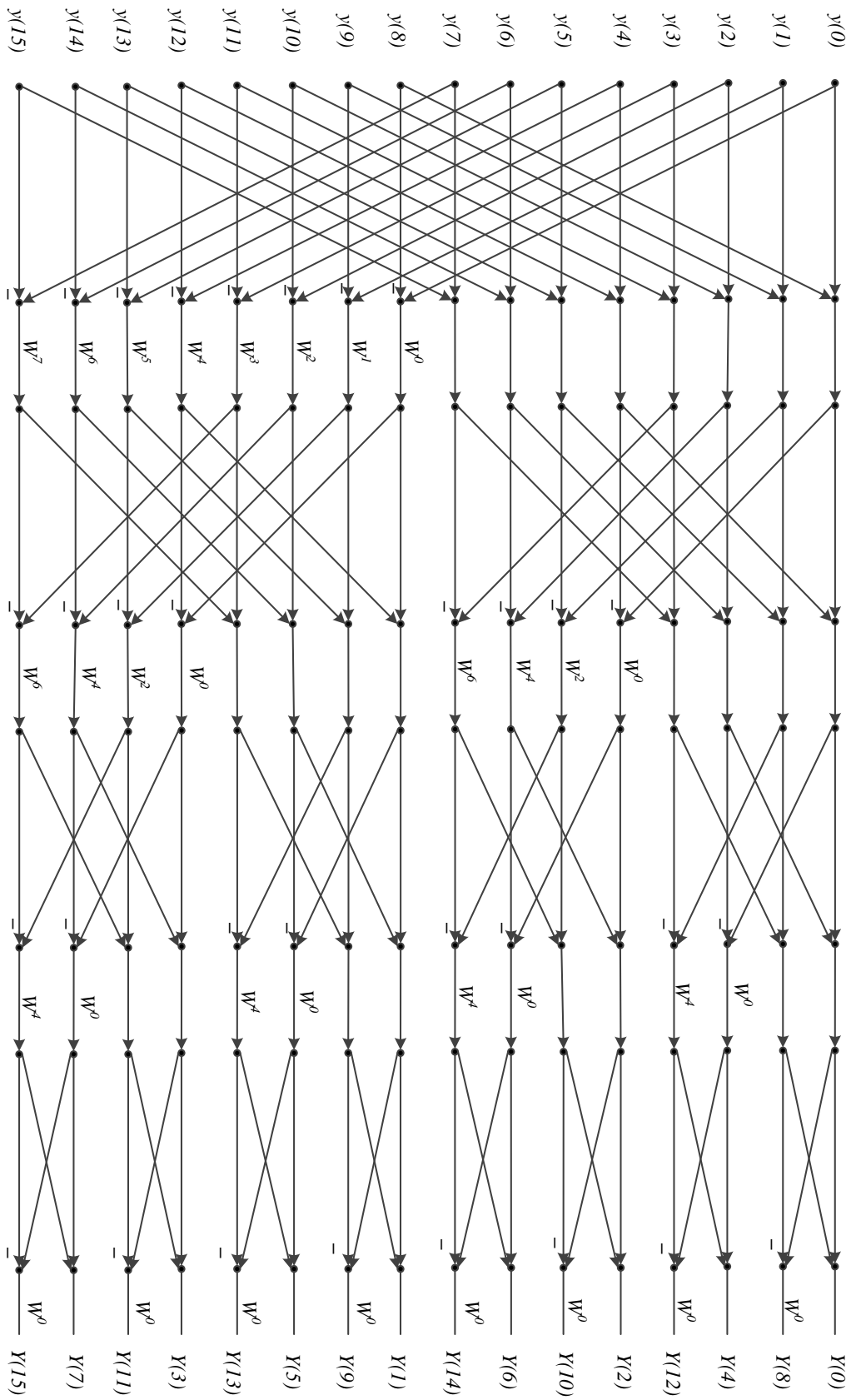


圖 十一、十六點 FFT 硬體架構

而在進行硬體數值運算過程，可參考圖十二.FFT 運算說明；其中  $B$  路徑下方有一負號 (-)，代表作  $X$  資料減  $Y$  資料的運算， $W^n$  為 FFT 之係數，須注意此係數有實部 ( $W^n\_real$ ) 與虛部 ( $W^n\_imaginary$ ) 資料，在  $B$  結果須進行複數運算，運算過程中需要個別紀錄、運算實部與虛部資料， $W^n$  係數表如表三.所示，其 FFT 輸出 (fft\_d0~fft\_d15) 資料格式如圖十三.所示。

關於複數的乘法運算，運算中實部與虛部需交互做運算。若有兩複數相乘為

$(a + bj) \times (c + dj)$ ，則相乘之後的結果如下圖十二.所示。

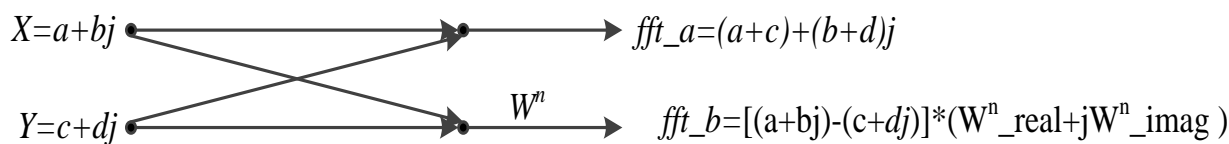
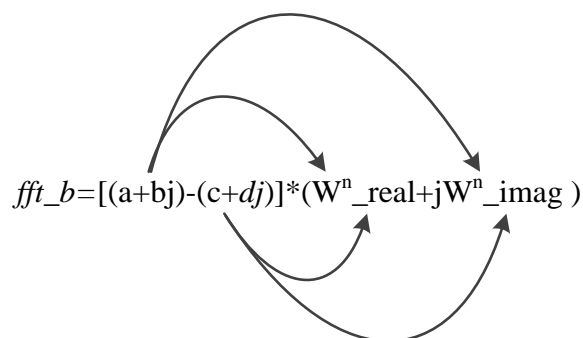


圖 十二、FFT 運算

$$fft\_a=(a+c)+(b+d)j$$

$$fft\_b=[(a+bj)-(c+dj)]*(W^n\_real+jW^n\_imag)$$



fft\_b 運算，進行交叉相乘，其運算式如下

$$fft\_b=(a-c)*W^n\_real+(a-c)W^n\_imagj+(b-d)W^n\_realj-(b-d)W^n\_imag$$

可整理為

$$fft\_b=(a-c)W^n\_real+(d-b)W^n\_imag+[(a-c)W^n\_imag+(b-d)W^n\_real]j$$

最後可整理為

實部資料	$(a-c)*W^n\_real+(d-b)*W^n\_imag$
虛部資料	$(a-c)*W^n\_imag+(b-d)*W^n\_real$

$W^n$			
$W^0$	$1.000 + 0.000j$	$W^4$	$0.000 - 1.0000j$
$W^1$	$0.923879532511287 - 0.382683432365090j$	$W^5$	$-0.382683432365090 - 0.923879532511287j$
$W^2$	$0.707106781186548 - 0.707106781186548j$	$W^6$	$-0.707106781186548 - 0.707106781186547j$
$W^3$	$0.382683432365090 - 0.923879532511287j$	$W^7$	$-0.923879532511287 - 0.382683432365089j$

表 3.  $W^n$ 係數

(表 3 內容，實部資料已存放於 Real\_Value\_Ref.dat，虛部資料已存放於 Imag\_Value\_Ref.dat)

正/負	實數整數	實數小數	正/負	虛部整數	虛部小數
1bit	7	8 bit	1 bit	7 bit	8 bit

圖 十三、FFT 資料格式 (fft\_d0 ~ fft\_d15)

而在進行 FFT 運算之前，需自行設計一個串列轉並列電路(Serial to Parallel)，如圖十一範例所示，使得資料依 FIR 濾波器輸出有效資料比對時序規格 (圖七) 輸入至 FFT 進行運算前，先將串列格式資料能夠轉成並列訊號 16 點處理一次，如圖十四範例所示，使得輸出資料能符合 FFT 輸出有效資料比對時序規格 (圖八)。關於 FFT 運算過程，參賽者可參考附錄 F. 資料，此資料為第一組測試樣本的第一筆 FFT 資料處理過程說明。

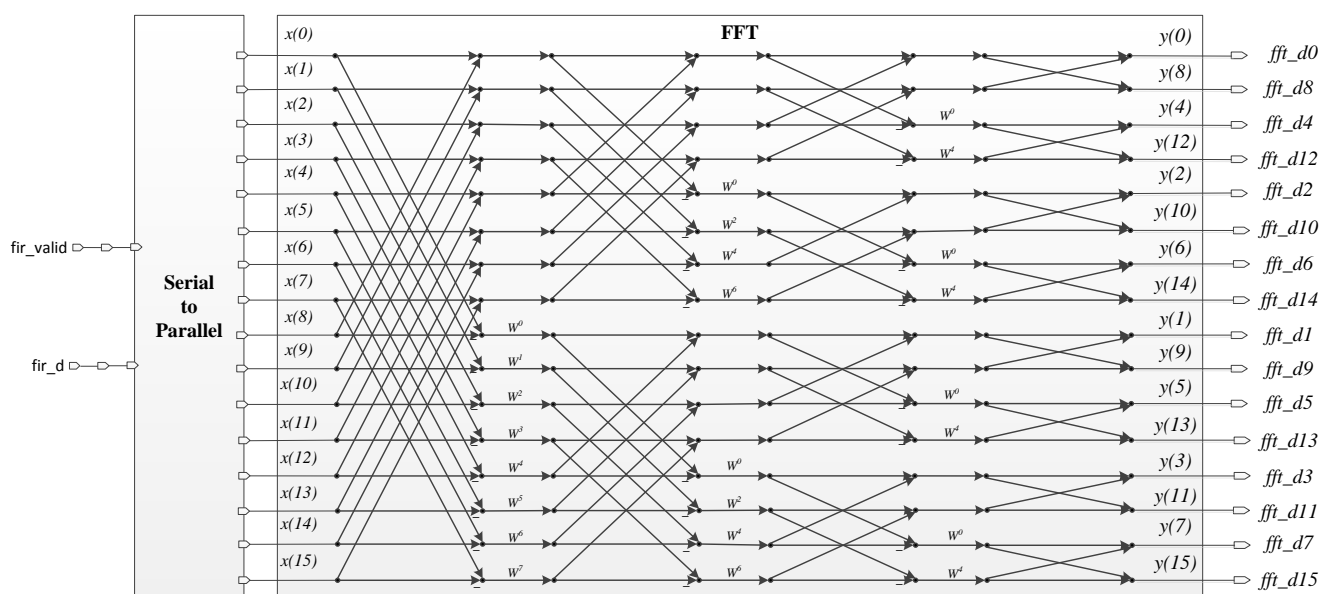


圖 十四、16 點快速傅利葉轉換 (FFT) 硬體架構含 Serial to Parallel 示意圖

## 2.7 頻譜分析電路 (Analysis)功能描述

將 FFT 運算結果輸入至頻譜分析電路後，須找出頻譜的主要頻段；本題目主要頻段定義為，FFT 運算結果所輸出的資料，會得到 16 個輸出結果 ( $Y(0)\sim Y(15)$ )，此 16 個輸出結果，實部資料與虛部資料各別平方後相加 ( $a+bj; a^2+b^2$ )，數值最大者輸出  $Y(n)$  的編號。

例如： $Y(0)=10$ 、 $Y(1)=15$ 、 $Y(2)=512\cdots Y(15)=20$ ，其中  $Y(2)$  的數值最大，故 freq 輸出 4'b0010。

### 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，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 E)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。本試題將提供兩組測試程式，每一功能均須通過此兩組測試程式方可視為通過；CIC 將在評分時新增一組測試向量，並以此共 3 組測試程式來評分。

CIC 將本試題依 RTL simulation 及電路合成結果區分為下面 3 個等級，以作為功能完成度之評分；若為同一等級則以檔案上傳時間(即作答時間的長短)來評分：

1. C 等級：測試樣本一至二 (testfixture1 & testfixture2) 通過 FIR filter 輸出資料比對
2. B 等級：測試樣本一至二 (testfixture1 & testfixture2) 通過 FIR filter 與 FFT 輸出資料比對
3. A 等級：測試樣本一至二 (testfixture1 & testfixture2) 通過 FIR filter、FFT 與 Analysis 輸出結果比對

請注意，我們將以各參賽隊伍的設計結果正確為前提，並以最後上傳檔案版本及時間為評分依據。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計分。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 為快速傅利葉轉換公式推導；附錄 D 為評分用檔案，亦即參賽者必須回傳至 CIC 的檔案資料；附錄 E 則為設計檔上傳步驟說明。

### 附錄A 軟體環境

競賽中所提供的設計軟體環境與版本如下表六。驗證評分時，係以所列軟體及版本作為驗證依據。

表 六、設計軟體版本

Functionality	Corresponding EDA tools
Logic Simulator	nc-verilog v11.10 modelsim v10.2 vcs v2011.12-sp1
Logic Synthesizer	design-compiler v2012.06-sp5

**注意！** 評分時將以nc-verilog v11.10版的結果為主。



## 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

表七、設計檔

檔名	說明
testfixture1.v testfixture2.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
FAS.v (FAS.vhd)	參賽者所使用的設計檔，已包含系統輸/出入埠之宣告
FIR_coefficient.dat	FIR filter 之係數檔，參賽者可自行決定是否使用。 本內容所顯示之資料格式為 20 bits。其中 MSB 為前 4 bits 表示整數位；LSB 為後 16 bits 表示小數位。
Golden1_FIR.dat Golden2_FIR.dat	兩組測試樣本的 Golden Pattern。每一樣本提供 FIR Filter 運算後的 golden pattern，各有 1024 筆資料需做比對。[註 1.][註 2.]
Golden1_FFT_real.dat Golden2_FFT_real.dat	兩組測試樣本的 Golden Pattern。每一樣本提供 FFT 運算後的實部結果 golden pattern，各有 1024 筆資料需做比對。[註 1.][註 2.]
Golden1_FFT_imag.dat Golden2_FFT_imag.dat	兩組測試樣本的 Golden Pattern。每一樣本提供 FFT 運算後的虛部結果 golden pattern，各有 1024 筆資料需做比對。[註 1.][註 2.]
Real_Value_Ref.dat	$w^n$ 的實部數值檔，參賽者可自行決定是否使用。 本內容所顯示之資料格式為 32 bits。其中 MSB 為前 16 bits 表示整數位；LSB 為後 16 bits 表示小數位。
Imag_Value_Ref.dat	$w^n$ 的虛部數值檔，參賽者可自行決定是否使用。 本內容所顯示之資料格式為 32 bits。其中 MSB 為前 16 bits 表示整數位；LSB 為後 16 bits 表示小數位。
synopsys_dc.setup	Design Compiler 初始設定範例檔案
FAS.sdc	Design Compiler 電路合成規範檔
report.000	結果報告範本

[註 1.] 兩組 golden pattern 皆以十六進制表示。

[註 2.] FFT 的 Golden pattern 均有設定一定程度的容許誤差。只要於每次 FFT 乘法運算後，小數點後的 bit 數留下 16 bits 以上，即可讓最終結果落在容許誤差範圍內。

2. 請使用 **FAS.v(vhd)**，進行本題目之設計。其模組名稱、輸出/入埠宣告如下所示：

### **FAS.v**

```
module FAS (data_valid, data, clk, rst, fir_d, fir_valid, fft_valid, done, freq,
            fft_d0, fft_d1, fft_d2, fft_d3, fft_d4, fft_d5, fft_d6, fft_d7, fft_d8,
            fft_d9, fft_d10, fft_d11, fft_d12, fft_d13, fft_d14, fft_d15);

input clk, rst;
input data_valid;
input [15:0] data;

output fir_valid, fft_valid;
output [15:0] fir_d;
output [31:0] fft_d0, fft_d1, fft_d2, fft_d3, fft_d4, fft_d5, fft_d6, fft_d7, fft_d8;
output [31:0] fft_d9, fft_d10, fft_d11, fft_d12, fft_d13, fft_d14, fft_d15;
output done;
output [3:0] freq;
//`include "../dat/FIR_coefficient.dat"

endmodule
```

### **FAS.vhd**

```
FAS.vhd
Library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;
ENTITY FFA IS Port(
clk : in std_logic;
rst : in std_logic;
data_valid : in std_logic;
fir_valid : out std_logic;
fft_valid : out std_logic;
done : out std_logic;
data : in std_logic_vector(15 downto 0);
fir_d : out std_logic_vector(15 downto 0);
fft_d0 : out std_logic_vector(31 downto 0);
fft_d1 : out std_logic_vector(31 downto 0);
fft_d2 : out std_logic_vector(31 downto 0);
fft_d3 : out std_logic_vector(31 downto 0);
fft_d4 : out std_logic_vector(31 downto 0);
fft_d5 : out std_logic_vector(31 downto 0);
fft_d6 : out std_logic_vector(31 downto 0);
fft_d7 : out std_logic_vector(31 downto 0);
fft_d8 : out std_logic_vector(31 downto 0);
fft_d9 : out std_logic_vector(31 downto 0);
fft_d10 : out std_logic_vector(31 downto 0);
fft_d11 : out std_logic_vector(31 downto 0);
fft_d12 : out std_logic_vector(31 downto 0);
fft_d13 : out std_logic_vector(31 downto 0);
fft_d14 : out std_logic_vector(31 downto 0);
fft_d15 : out std_logic_vector(31 downto 0);
freq : out std_logic_vector(4 downto 0);
);
```

```
END FAS;  
ARCHITECTURE FAS_arc OF FAS IS  
BEGIN  
END FAS_arc;
```

3. 比賽共提供兩組測試樣本，兩個測試程式均已各自對應到各 golden pattern 了，參賽者只需要注意這些檔案的路徑即可。參賽者可依下面範例來進行模擬：

- ncverilog 指令範例如下：  
*ncverilog testfixture1.v FAS.v*  
*或 ncverilog testfixture2.v FAS.v*
- 若使用 modelsim，則是在 compiler verilog 時，使用下面指令：  
*vlog testfixture1.v*  
*或 vlog testfixture2.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*

4. 因波形檔很大，請以 fsdb dump 取代 vcd dump，dump fsdb 須使用指令如下：

- ncverilog 指令範例如下：  
*ncverilog testfixture1.v FAS.v +access+r*
- modelsim 使用者，請直接使用內建波形來進行除錯。

## 附錄 C 快速傅利葉轉換公式推導

由於原始的離散型傅利葉轉換，需要用到大量的乘法器（MUL）及加法器（ADD），其 DFT 與 FFT 乘法器、加法器個數如表八所示，所以一般會將離散型傅利葉公式一拆解成公式三。

表八 DFT 與 FFT 運算元個數

DFT		FFT	
MUL	ADD	MUL	ADD
$(N-2)^2$	$N^2$	$\frac{N}{2 \log_2 N - (N-1)}$	$N \log_2 N$

$$y(k) = \sum_{m=0}^{\frac{N}{2}-1} w^{mk} x(m) + w^{\frac{N}{2}k} \sum_{m=0}^{\frac{N}{2}-1} w^{mk} x(m + \frac{N}{2}) \quad \text{公式三}$$

透由尤拉公式（Euler Formula）可知  $e^{ix} = \cos x + i \sin x$ ，其中  $w^{\frac{N}{2}}$  對應到旋轉角度為 180 度，故此因子可被視為  $(-1)^k$  如公式四。

$$w^{\frac{N}{2}k} = (-1)^k \quad \text{公式四}$$

將此因子代入公式三，可以表示為公式五。

$$y(k) = \sum_{m=0}^{\frac{N}{2}-1} w^{mk} [x(m) + (-1)^k x(m + \frac{N}{2})] \quad \text{公式五}$$

將  $k$  拆為奇數部分與偶數部分，可得到公式六。

$$\begin{aligned} y(2r) &= \sum_{m=0}^{\frac{N}{2}-1} [x(m) + x(m + \frac{N}{2})] w^{m2r} \\ y(2r+1) &= \sum_{m=0}^{\frac{N}{2}-1} [x(m) - x(m + \frac{N}{2})] w^m \bullet w^{m2r} \\ r &= 0, 1, \dots, \frac{N}{2} - 1 \end{aligned} \quad \text{公式六}$$

並可將公式六代換表示為公式七

$$\begin{aligned} x'0(m) &= x(m) + x(m + \frac{N}{2}) \\ x'1(m) &= [x(m) - x(m + \frac{N}{2})] w^m \end{aligned} \quad \text{公式七}$$

最後，可推導出公式八，而在硬體實現上，架構如圖十一。

其 2 點及 4 點快速傅利葉轉換，硬體架構如圖十六所示。

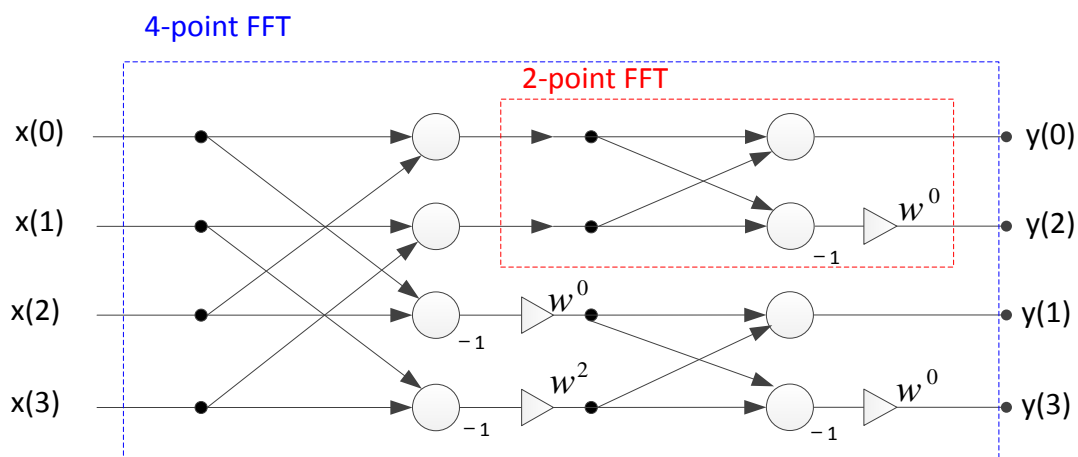


圖 十五、2 點及 4 點快速傅利葉轉換 (FFT) 硬體架構

其 8 點快速傅利葉轉換硬體架構如圖十七所示。

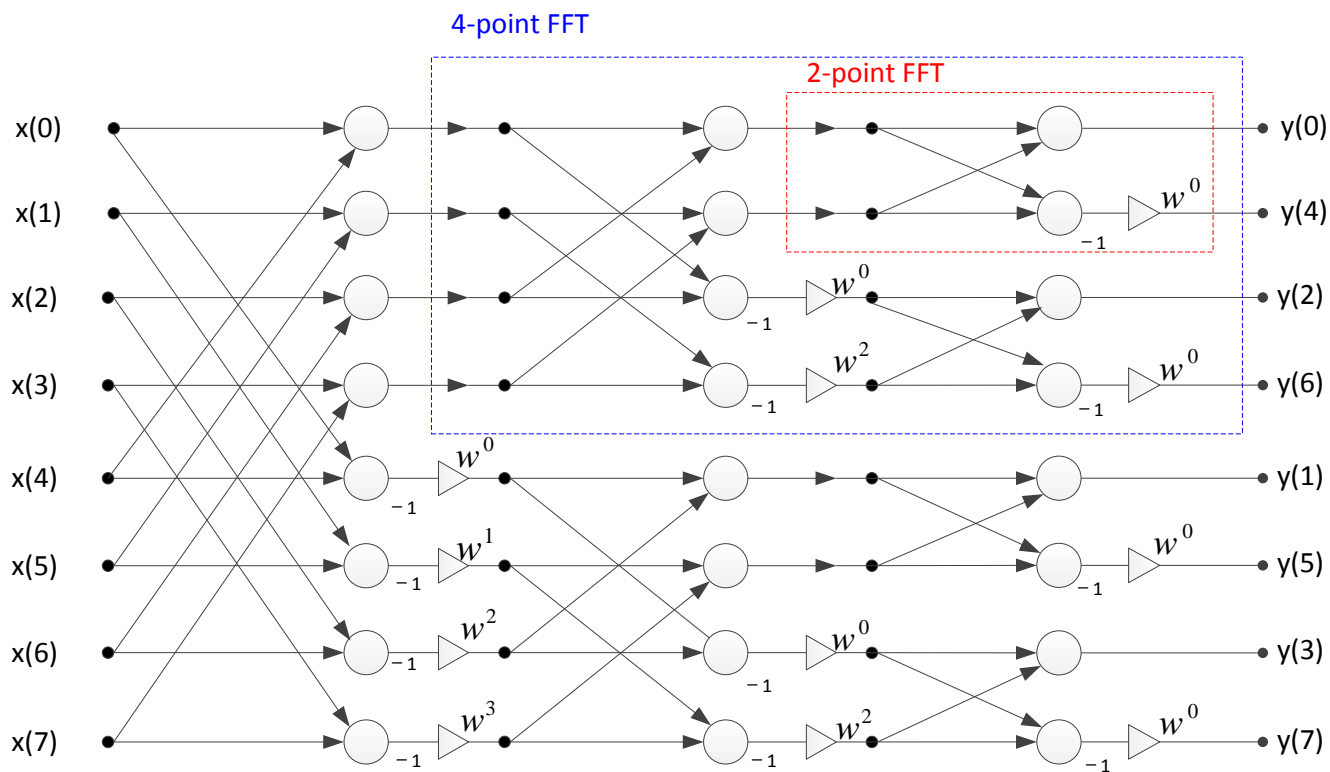


圖 十六、8 點快速傅利葉轉換 (FFT) 硬體架構

## 附錄 D 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 十七所示。(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 九、評分用檔案

<b>RTL category</b>		
<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
<b>Gate-Level category</b>		
<i>Design Stage</i>	<i>File</i>	<i>Description</i>
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

ftp 帳號(FTP 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): FAS.v

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

gate-level 檔案名稱(gate\_level filename): FAS\_syn.v

gate-level sdf filename: FAS\_syn.sdf

design compiler 合成資料庫(dc library): FAS\_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 十七、report.000 的範本

## 附錄 E 檔案上傳

所有包含於如附錄 D 中表格所示的檔案，均需要提交至 CIC。提交的設計檔案，需要經過壓縮於同一個資料夾下，並為\*.zip 或\*.tar 格式(建議以\*.tar 格式為優先)<sup>1</sup>。步驟如下：

1. 建立一個 result\_xxx 資料夾。其中“xxx”表示繳交版本(即第幾次上傳之編號)。例如“001”表示為第一次上傳；“002”表示為第二度上傳；003 表示為第三度上傳，以此類推…。
2. 參考附錄 D 評分用檔案，將所有繳交檔案複製到 result\_xxx 資料夾
3. 參賽者需依據設計內容編輯報告檔案“report.xxx”(如圖十四所示)，並變更其檔名。例如，設計的最初(第一次上傳)版本，報告檔名應命名為“report.001”並置於 result\_001 資料夾下。倘若參賽者於上傳設計後，若需再變更設計，則需進行第二次上傳以便更新(update)，此時檔名應改為“report.002”並置於 result\_002 資料夾下，再連同設計檔一併壓縮上傳。亦即 result\_xxx 與 report.xxx 之“xxx”編號需一致。而評審將以**最後上傳的設計檔及報告檔編號進行評分作業**。
4. 確認該提交的檔案均已備妥，即可將 result\_xxx 資料夾進行壓縮成\*.tar(或\*.zip)格式，如 result\_xxx.tar(或 result\_xxx.zip)。
5. 待完成壓縮，即可進行 FTP 上傳 (CIC 已於競賽當日的前 4 天，將 FTP 的 username 及 password 藉電子郵件，寄送到參賽者信箱)。倘若上傳設計檔過程中，有任何問題，請與 CIC 接洽。

**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)

6. 若你需要繳交更新版本，請重覆以上步驟，並記得修改 report 檔及壓縮檔的版本編號，因為你無法修改或刪除或覆蓋之前上傳的資料。再次提醒各參賽隊伍，result\_xxx 目錄名稱與 report.xxx 報告檔之“xxx”編號需一致，且依各隊伍上傳設計檔之次數進行編號。評審將以最後上傳的設計檔及報告檔編號進行評分作業!!
7. 建議各參賽隊伍，無論是否完成設計，均需上傳設計檔案!

<sup>1</sup>假如參賽者無任何壓縮軟體，您可以至以下連結，下載壓縮軟體試用版。

A. winzip: <http://www.winzip.com>

B. 7zip: <http://www.7-zip.org/download.html>

附錄 F 16 點 FFT 運算過程說明

為方便除錯，表九提供輸入前 16 筆資料進行 FFT 運算之結果。

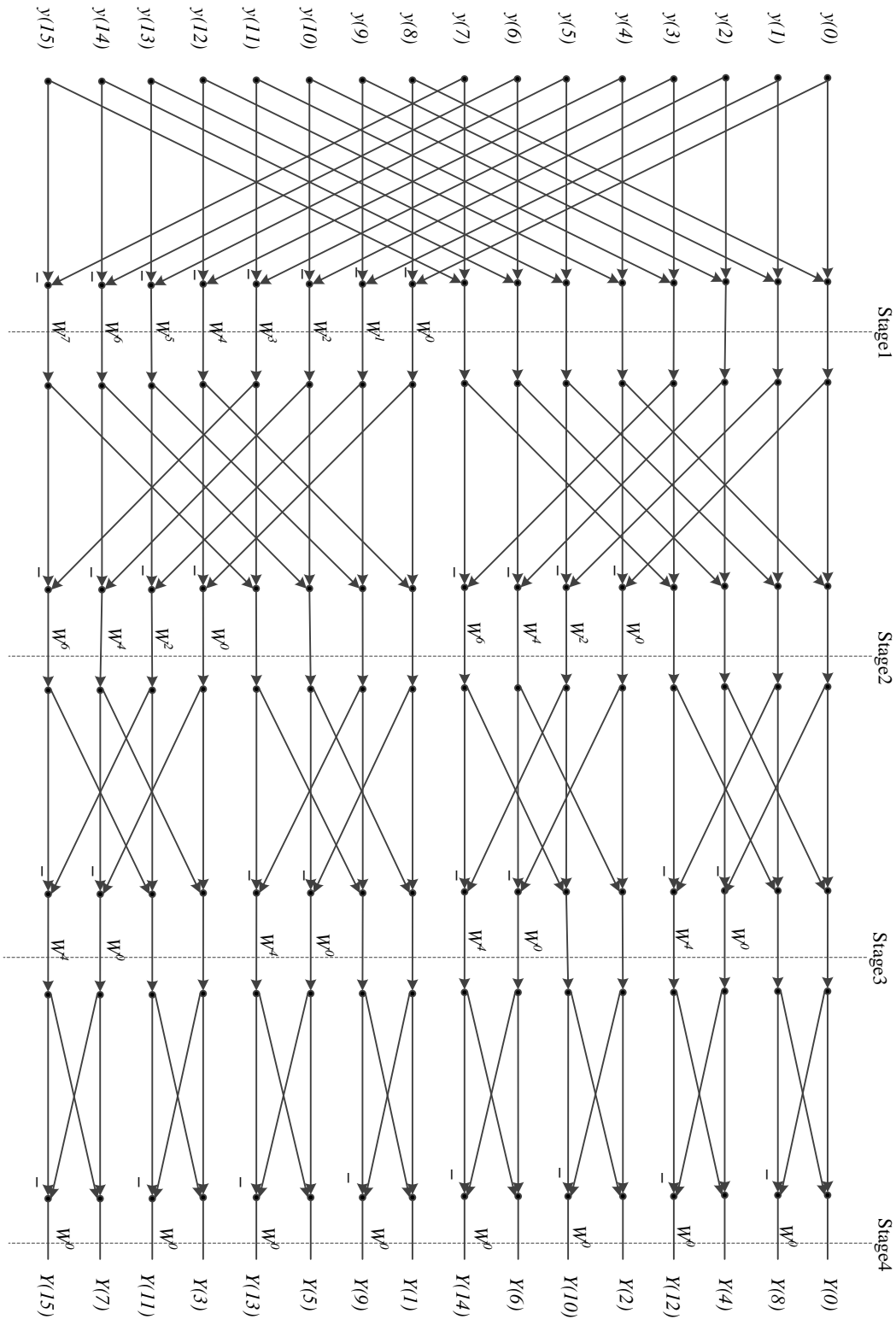


圖 十八、16 點 FFT 運算說明



表九、FFT 運算結果

Stage1		Stage2		Stage3	
實數	虛數	實數	虛數	實數	虛數
32'hFFFFDA00	32'h00000000	32'hFFFFFE00	32'h00000000	32'h00002D00	32'h00000000
32'hFFFFEA00	32'h00000000	32'h00001900	32'h00000000	32'h00005800	32'h00000000
32'hFFFFFE00	32'h00000000	32'h00002F00	32'h00000000	32'hFFFFCF00	32'h00000000
32'h00001300	32'h00000000	32'h00003F00	32'h00000000	32'h00000000	32'h00002600
32'h00002400	32'h00000000	32'hFFFFB600	32'h00000000	32'hFFFFB600	32'h00003300
32'h00002F00	32'h00000000	32'hFFFFCF36	32'h000030CA	32'hFFFFE0E3	32'h00004277
32'h00003100	32'h00000000	32'h00000000	32'h00003300	32'hFFFFB600	32'hFFFFCD00
32'h00002C00	32'h00000000	32'h000011AD	32'h000011AD	32'h00001F1D	32'h00004277
32'h00014400	32'h00000000	32'h00014400	32'hFFFEA000	32'h0002BE4A	32'hFFFD518C
32'h0001A370	32'hFFFF5244	32'h00016079	32'hFFFE097	32'h0002EA48	32'hFFFD5813
32'h0001645F	32'hFFFE9BA1	32'h00017A4A	32'hFFFE18C	32'hFFFFC9B6	32'hFFFFEE74
32'h0000B379	32'hFFFE4EB5	32'h000189CF	32'hFFFEA77C	32'h0000091B	32'h00002956
32'h00000000	32'hFFFEA000	32'h00014400	32'h00016000	32'hFFFFC9B6	32'h0000118C
32'hFFFFBD09	32'hFFFF5E53	32'h00014F67	32'hFFFE9F8B	32'hFFFFF6E7	32'h00002958
32'h000015EB	32'h000015EB	32'hFFFE85B6	32'hFFFE18C	32'h0002BE4A	32'h0002AE74
32'h0000D656	32'h000058C7	32'hFFFEA780	32'h000189CD	32'hFFFD15BE	32'hFFFD5819

Stage4	
實數	虛數
32'h00008500	32'h00000000
32'hFFFFD500	32'h00000000
32'hFFFFCF00	32'h00002600
32'hFFFFCF00	32'hFFFFDA00
32'hFFFF96E3	32'h00007577
32'hFFFFD51D	32'hFFFFF089
32'hFFFFD51D	32'h00000F77
32'hFFFF96E3	32'hFFFF8A89
32'h0005A892	32'hFFFAA99F
32'hFFFFD402	32'hFFFFF979
32'hFFFFD2D1	32'h000017CA
32'hFFFFC09B	32'hFFFFC51E
32'hFFFFC09D	32'h00003AE4
32'hFFFFD2CF	32'hFFFFE834
32'hFFFFD408	32'h0000068D

32'h0005A88C	32'h0005565B
--------------	--------------