

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 E.icdesign.ncu@gmail.com

信件主旨:登錄(ID: E?????)(請填上自己的報名 ID)

信件內容:

組別: E 組

ID: E????? (例: E00001)

姓名: 李大華、王小明

2013 IC Design Contest Preliminary

大學組可程式邏輯(FPGA)設計組

Frequency Analysis System

1. 問題描述

本題目須完成 2 個獨立電路模組，分別為 2 點 FFT(FFT_PE)電路模組及頻譜分析系統(FAS)電路模組。

FAS 電路模組包含了緩衝電路 (Buffer)、16 點快速傅利葉轉換 (FFT: Fast Fourier Transform) 及頻譜分析電路 (Analysis)，架構如圖一所示。2 點 FFT (FFT_PE) 電路模組是為了讓設計者用來先行驗證快速傅利葉轉換電路之複數 (complex number) 運算，架構如圖二所示，若此模組通過驗證 (參考 3. 評分標準) 則有 A 等級分數。請先完成 2 點 FFT (FFT_PE) 電路模組，再擴充至 16 點 FFT (FFT_16_P) 電路模組，進而完成整個頻譜分析系統 (FAS) 電路。

FAS 模組有 4 只信號輸入 (clk, rst, data, data_valid)、19 只信號輸出 (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, 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。

FFT_PE 模組有 6 只信號輸入 (clk, rst, a, b, ab_valid, power)、3 只信號輸出 (fft_a, fft_b, fft_pe_valid)，如圖二所示，關於各輸入/輸出信號的功能說明，請參考表二。

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。
3. 本次設計分為兩個獨立設計檔（**top module**），個別評分（**FFT_PE.v**、**FAS.v**）

2.1 系統方塊圖

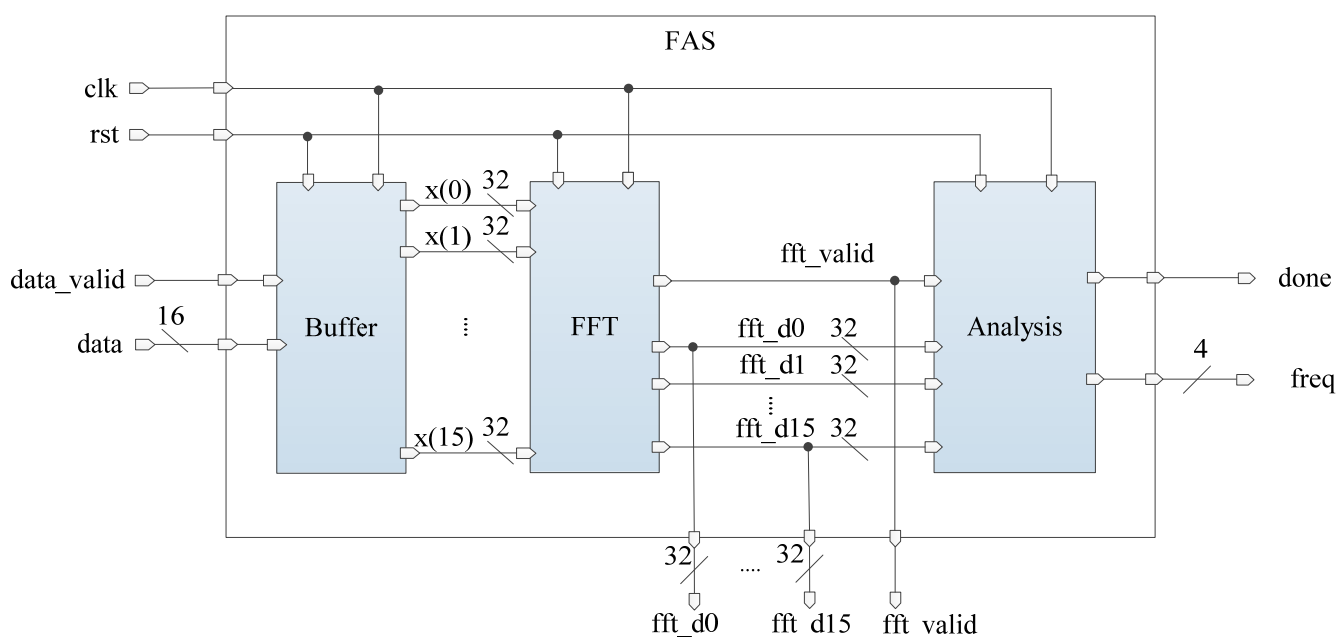


圖 一、系統方塊圖

2.1.1 2 點 FFT 方塊圖

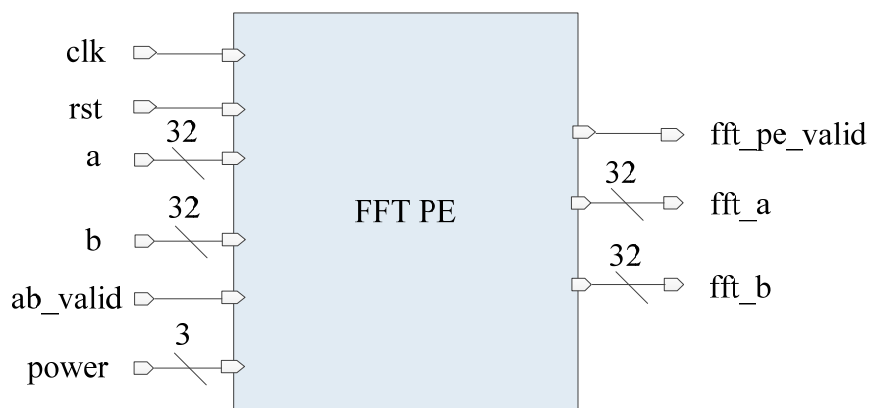


圖 二、2 點 FFT 方塊圖

2.2 輸入/輸出介面

表 一、FAS 模組輸入/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	資料有效信號。 說明：當主控端傳送資料時，會將此信號設為 high。
<i>data</i>	input	16	主控端資料訊號輸入。
<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>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。(請參考 page 11, 2.7 該段說明)

表 二、FFT_PE 模組輸入/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>ab_valid</i>	input	1	資料預備信號。 說明：當主控端傳送資料時，會將此信號設為 high。
<i>a</i>	input	32	主控端資料訊號輸入。
<i>b</i>	input	32	主控端資料訊號輸入。
<i>fft_a</i>	output	32	32 位元，2 點 FFT 資料輸出訊號。
<i>fft_b</i>	output	32	32 位元，2 點 FFT 資料輸出訊號。
<i>fft_pe_valid</i>	output	1	FFT 資料有效信號。
<i>power</i>	input	3	選擇 W^n 係數信號。

2.3 系統功能描述

在進行 16 點 FFT 運算之前，需自行設計一個緩衝電路(Buffer)，先將每一筆 16 bits 的資料緩衝至 16 筆後，再輸入至 FFT 模組運算一次，使得輸出資料能符合 FFT 輸出有效資料比對時序規格。

FAS 一開始輸入之資料格式為 16({8 bit 整數,8bit 小數})，而在負數表示方面，採用二補數表示之。

2.3.1 複數運算說明

FFT 在運算過程中的 W^n 係數為複數 (complex number)，包含實數與虛數數值，在電路的處理過程中，可當作一般資料個別運算 (實數與實數運算、虛數與虛數運算)，但在作乘法運算時，須注意交叉相乘時的運算，例如： $(a+bj) * (c+dj) = ac+adj+bcj-bd$ ，整理後為 $(ac-bd)+(ad+bc)j$ 。

2.4 時序規格圖及資料格式

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、2 點 FFT 輸出有效資料比對時序規格、16 點 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三所示，其中的時脈週期 t_{CYCLE} 預設為 20ns。done 信號運算程過中，都設定為 low，當 FAS 每完成一次 16 點的 FFT 運算時，將 done 信號設定為 high，此時測試程式將比對 freq 信號輸出之資料 (請參考 page 11，2.7 該段說明)，本系統將由主控端輸入 160 筆資料，故會有 10 次運算結果。

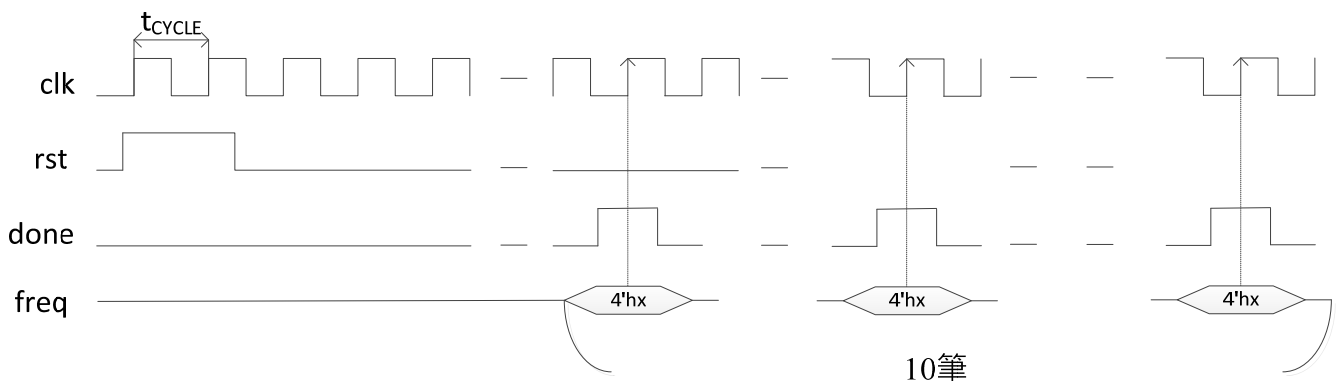


圖 三、主控端時脈、重置與完成訊號時序圖

2.4.2 主控端資料傳輸時序規格

● FAS 資料傳輸時序規格說明

從主控端 (host) 傳送的測試檔之資料共有 160 筆的序列資料，每一個時脈會送出一筆資料，每一筆資料都是實數 (real number)，資料位元寬度為 16 bit (前 8bit 為整數，後 8bit 為小數)，在運算過程中，會有複數運算。其 FAS 資料傳輸時序規格如圖四所示，資料格式如圖五。

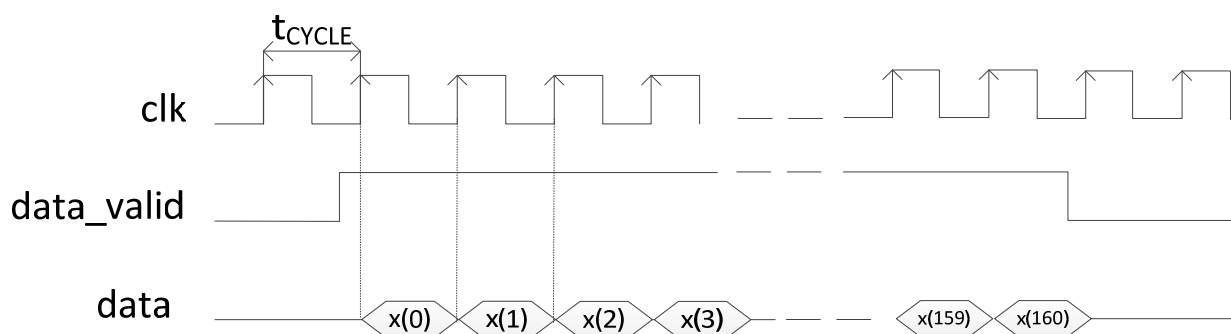


圖 四、FAS 主控端資料傳輸時序圖

data	
整數	小數
8bit	8 bit

圖 五、FAS 輸入資料格式 (Port:data)

● FFT_PE 資料傳輸時序規格說明

2 點 FFT 之測試檔案，提供資料給 a,b 輸入埠各 8 筆，當主控端將 ab_valid 訊號設定為 high 時，每一個時脈送出一筆資料，故會有 10 次運算結果輸出，實部資料與虛部資料位元寬度各為 16 bit (前 8bit 為實數，後 8 bit 為虛數)。其時序規格如圖六所示，其資料格式如圖七。

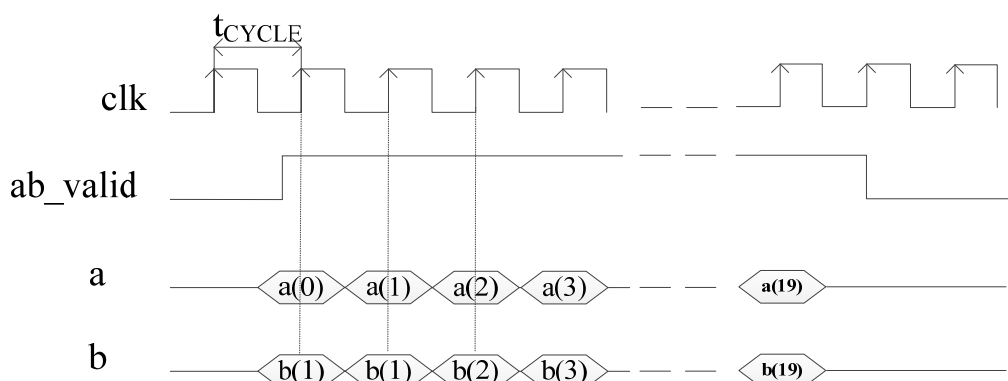


圖 六、FFT_PE 主控端資料傳輸時序圖

a, b			
實部資料格式		虛部資料格式	
整數	小數	整數	小數
8 bit	8 bit	8 bit	8 bit

圖 七、FFT_PE 輸入資料格式

2.4.3 2 點 FFT 輸出有效資料比對時序規格

測試檔將在 fft_pe_valid 信號設定為 high 時進行資料比對，其比對時序如圖八與資料格式如圖九。

Hint:在每一級 FFT_PE 運算過程中為確保精確度，實數、虛數進行運算時，最少各採用 32 位元（{整數 16 bit,小數 16bit}）精確度表示，待運算完後，取其 16 位元（{整數 8bit,小數 8 bit}）表示。

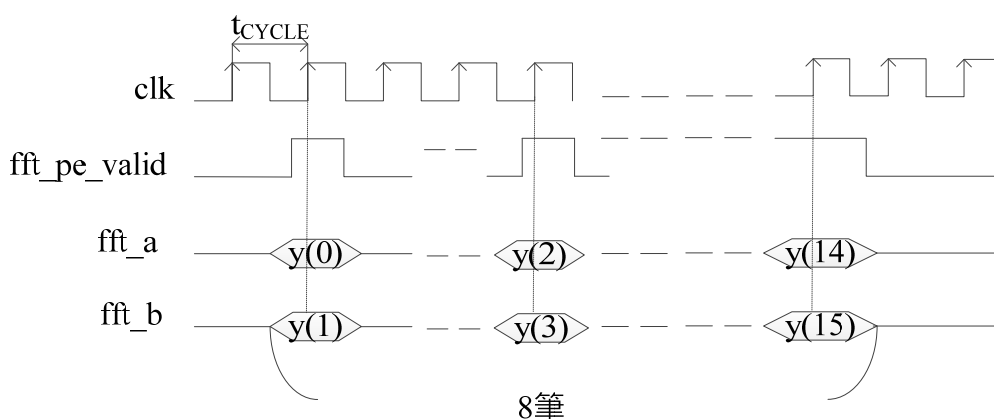


圖 八、FFT_PE 資料比對格式

fft_a, fft_b			
實部資料格式		虛部資料格式	
整數	小數	整數	小數
8 bit	8 bit	8 bit	8 bit

圖 九、FFT_PE 輸出資料格式

2.4.4 16 點 FFT 輸出有效資料比對時序規格

測試檔將在 fft_valid 信號設定為 high 時進行資料比對，其比對時序如圖十。fft_d0~fft_d15 輸出時，實部資料與虛部資料只取其 16 bit，如圖十一所示資料格式。

Hint:在每一級 FFT_PE 運算過程中為確保精確度，實數、虛數進行運算時，最少各採用 32 位元（{整數 16 bit,小數 16bit}）精確度表示，待運算完後，取其 16 位元（{整數 8bit,小數 8 bit}）表示。

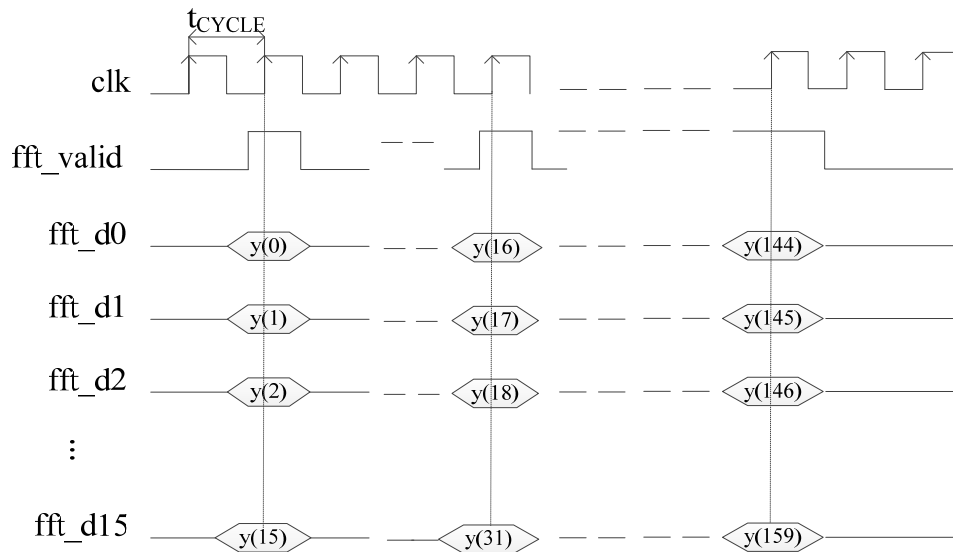


圖 十、FFT 端資料比對時序圖

fft_d0~fft_d15			
實部資料格式		虛部資料格式	
整數	小數	整數	小數
8 bit	8 bit	8 bit	8 bit

圖 十一、FFT 輸出資料格式

2.5 2 點快速傅利葉轉換 (2 Point FFT)功能描述

圖十二為 2 點 FFT 是為了解設計者用來簡單驗證其電路之複數(complex number)運算結果，其係數請使用表三之 W^0 。

圖十二中 Y 路徑下方有一負號 (-)，代表作 Y 資料的二補數運算， W^n 為 FFT 之係數，此係數為複數代表有實部 (W^n_real) 與虛部 (W^n_imag) 資料，實部資料與虛部資料各別運算、儲存。進行進行複數運算後，將結果輸出至 fft_a 、 fft_b ，並依照 FFT_PE 資料格式輸出。

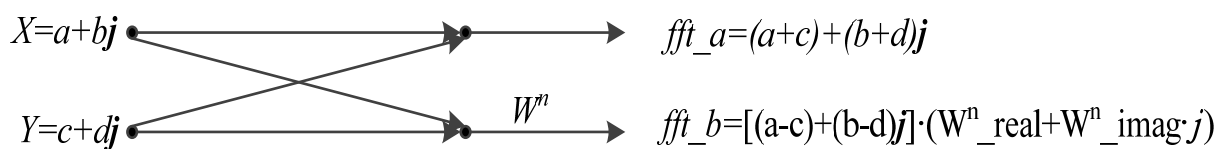


圖 十二、FFT_PE 硬體架構

$$fft_a = (a+c) + (b+d)j$$

$$fft_b = [(a+bj) - (c+dj)] * (W^n_real + W^n_imagj) \text{ (須作交叉相乘，如圖十三)}$$

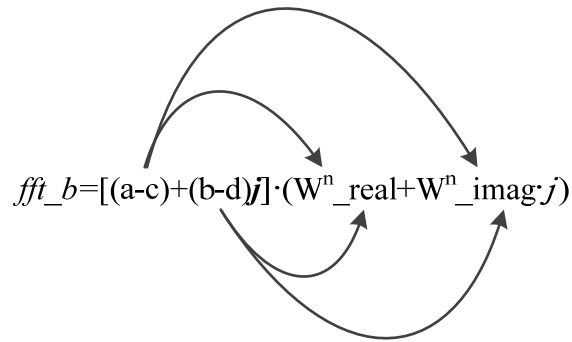


圖 十三、fft_b 交叉相乘運算

fft_b 運算，進行交叉相乘，其運算式如下：

$$\text{fft_b} = (a-c)W^n_real + (a-c)W^n_imagj + (b-d)W^n_realj - (b-d)W^n_imag$$

$$\text{fft_b} = (a-c)W^n_real + (d-b)W^n_imag + [(a-c)W^n_imag + (b-d)W^n_real]j$$

最後可整理為

實部資料	$(a-c)*W^n_real + (d-b)*W^n_imag$
虛部資料	$(a-c)*W^n_imag + (b-d)*W^n_real$

圖 十四、fft_b 輸出運算式整理 (W^n_real 為實部係數， W^n_imag 為虛部係數)

2.6 16 點快速傅利葉轉換 (FFT：Fast Fourier transform) 功能描述

本 FAS 系統中所使用的快速傅利葉轉換 (FFT)，為 16 點的快速傅利葉轉換，硬體架構如圖十五所示， W^n 係數如表三所示。關於 16 點 FFT 運算過程說明，可參考附錄 F，圖二十。

表 三、 W^n 係數

W^n					
係數	實部 (W^n_real)	虛部 (W^n_imag)	係數	實部 (W^n_real)	虛部 (W^n_imag)
W^0	32'h00010000	32'h00000000	W^4	32'h00000000	32'hFFFF0000
W^1	32'h0000EC83	32'hFFFF9E09	W^5	32'hFFFF9E09	32'hFFFF137D
W^2	32'h0000B504	32'hFFFF4AFC	W^6	32'hFFFF4AFC	32'hFFFF4AFC
W^3	32'h000061F7	32'hFFFF137D	W^7	32'hFFFF137D	32'hFFFF9E09

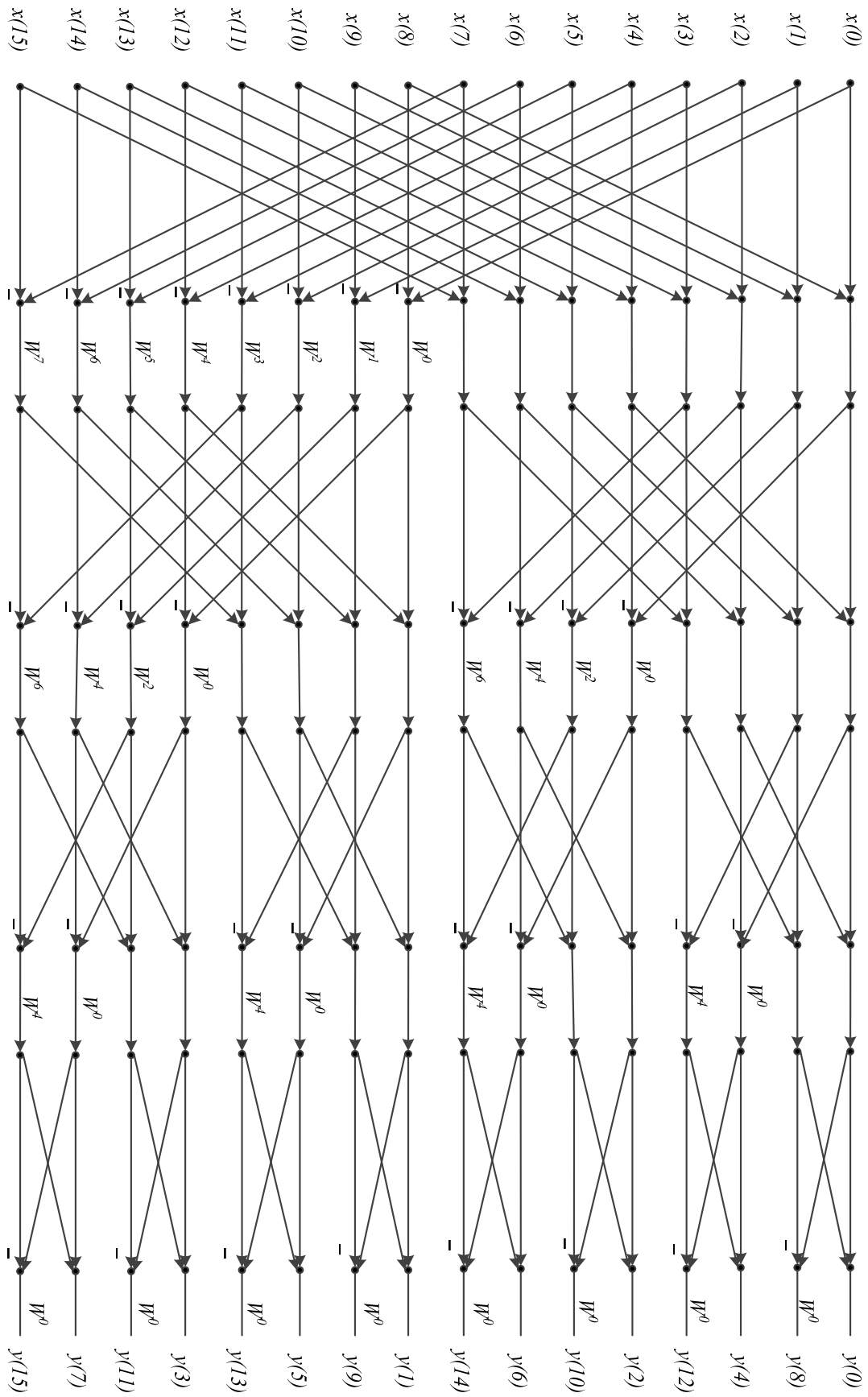


圖 十五、16 點 FFT 硬體架構

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 (不會有相同結果，只有一個最大值)。

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。

2.8 FPGA 元件指定

本次 FPGA 組初賽主要由 Mentor Graphics 公司 Modelsim 進行 RTL 模擬來進行評分，唯各參賽者所設計之 FIR Filter、FFT、Analysis 電路必須可分別在 FPGA 開發工具進行電路合成。請各參賽者依使用 Altera 或是 Xilinx FPGA 開發工具之不同，選擇如下的元件及對應之 FPGA 發展工具進行設計編譯。請參賽者注意，**本試題第三節所述的評分標準中，關於元件資源使用率(usage ratio)及設計效能，或選用不同之 FPGA 開發工具，並不會影響本競賽初賽的評分!! 且各位參賽者需在 report file 中填寫所使用之合成器為 Quartus II 或是 ISE。**本次競賽所使用之 FPGA 開發環境軟體版本以附錄 A 中所指定為標準。

表四、Altera 組元件指定

Altera	
Target Family	Cyclone IV E
Target Device	EP4C30
Target Package	FBGA
Pin count	484
Target Speed	8

表五、Xilinx 組元件指定

Xilinx	
Target Family	Spartan 6
Target Device	XC6SLX25
Target Package	FG (FBGA)
Pin count	484
Target Speed	-2

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。

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. A 等級：測試樣本 (testfixture0) 通過 2 點 FFT 輸出資料比對
2. B 等級：已通過 A 等級，並以測試樣本一至二 (testfixture1 & testfixture2) 通過 16 點 FFT 輸出資料比對
3. C 等級：已通過 A 等級與 B 等級，並以測試樣本一至二 (testfixture1 & testfixture2) 通過 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 F 為 16 點 FFT 運算過程說明。

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
Design Entry	VHDL/Verilog
FPGA Synthesizer	Synopsys Synplify Premier (2012.09-SP1) 、 Mentor Graphics Precision Synthesizer (PS2012b)
FPGA Simulator	Mentor Graphics ModelSim SE (v10.2)
FPGA tools	Altera QuartusII (v12.1) 、Xilinx ISE (v14.2i)

各隊伍可以利用上述 EDA 軟體工具來檢查自己設計的正確性及設計是否可合成，也可直接使用 FPGA 廠商的發展軟體(如 Altera Quartus II 或是 Xilinx ISE)中內建的編輯功能來完成這些動作，這些並不會影響評分的進行(請參考本試題卷第 3 節之評分標準說明)。**主辦單位是以(1)ModelSim 模擬結果及(2)電路是否可在 ISE/Quartus II 中合成作為評分的依據。**

如參賽手冊中刊載，採用之 ModelSim 為 CIC 提供學術界申請使用的 Mentor Graphics 公司發行之正式版，有支援 Verilog/VHDL 混合模擬的功能，若參賽者使用自行由 Altera/Xilinx 公司下載之 OEM 版 ModelSim 而無法順利利用主辦單位提供的 verilog testbench 進行模擬者，不在主辦單位負責範圍內。

附錄 B 設計檔案說明

1. 下表七、表八為主辦單位所提供各參賽者的設計檔案

表七、資料夾 FAS 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（testfixture1.v、testfixture2.v 為 FAS.v 測試檔）
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
Golden1_FFT_real.dat Golden2_FFT_real.dat	FAS 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 160 筆資料需做比對。 註：兩組 golden pattern 皆以十六進制表示。
Golden1_FFT_imag.dat Golden2_FFT_imag.dat	FAS 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 160 筆資料需做比對。 註：兩組 golden pattern 皆以十六進制表示。
Golden1_FIR.dat	FAS 測試樣本一輸入檔案。
Golden2_FIR.dat	FAS 測試樣本二輸入檔案。
STAGE_x_real.dat	Stage1~Stage4 實部比對檔案，供參賽者參考。
STAGE_x_imag.dat	Stage1~Stage4 虛部比對檔案，供參賽者參考。
Real_Value_Ref.dat	w^n 的實部數值檔，參賽者可自行決定是否使用
Imag_Value_Ref.dat	w^n 的虛部數值檔，參賽者可自行決定是否使用
report.000	結果報告範本

表八、資料夾 FFT_PE 設計檔

檔名	說明
testfixture0.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（testfixture0.v 為 FFT_PE.v 測試檔）
FFT_PE.v(FFT_PE.vhd)	參賽者所使用的設計檔，已包含系統輸/出入埠之宣告
PE_FFT_PATTERN_real.dat	FFT_PE 測試樣本輸入實部檔案。
PE_FFT_PATTERN_imag.dat	FFT_PE 測試樣本輸入虛部檔案。
PE_FFT_EXPECTED_real.dat	FFT_PE 一組測試樣本的 Golden Pattern 實部檔案。
PE_FFT_EXPECTED_imag.dat	FFT_PE 一組測試樣本的 Golden Pattern 虛部檔案。
W_real.dat	w^0 的實部數值檔，參賽者可自行決定是否使用

W_imag.dat	w^0 的虛部數值檔，參賽者可自行決定是否使用
report.000	結果報告範本

2. 請使用 **FAS.v(vhd)**，進行頻譜分析電路之設計。其模組名稱、輸出/入埠宣告如下所示：

FAS.v

```

module FAS(
    clk,
    rst,
    data_valid,
    data,
    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,
    fft_valid,
    done,
    freq
);

input  clk;
input  rst;
input  data_valid;
input signed [15:0] data;
output [31:0] 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;
output fft_valid;
output done;
output [3:0] freq;

endmodule

```

FFT_PE.v

```

module FFT_PE(
    clk,
    rst,
    a,
    b,
    power,
    ab_valid,
    fft_a,
    fft_b,
    fft_pe_valid
);

input clk, rst;
input signed [31:0] a, b;
input [2:0] power;
input ab_valid;
output [31:0] fft_a, fft_b;
output fft_pe_valid;

endmodule

```

FAS.vhd

```

Library IEEE;

```

```

use IEEE.std_Logic_1164.all;
use IEEE.numeric_std.all;
ENTITY FAS IS Port(
clk : in std_logic;
rst : in std_logic;
data_valid : in std_logic;
fft_valid : out std_logic;
done : out std_logic;

data : in 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(3 downto 0);
);

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

```

FFT_PE.vhd

```

Library IEEE;
use IEEE.std_Logic_1164.all;
use IEEE.numeric_std.all;
ENTITY FFT_PE IS Port(
clk : in std_logic;
rst : in std_logic;
ab_valid : in std_logic;
fft_pe_valid : out std_logic;

power : in std_logic_vector(2 downto 0);
a : in std_logic_vector(31 downto 0);
b : in std_logic_vector(31 downto 0);

fft_a : out std_logic_vector(31 downto 0);
fft_b : out std_logic_vector(31 downto 0);

);

END FFT_PE;
ARCHITECTURE FAS_arc OF FFT_PE IS
BEGIN

```


END FFT_PE_arc;

1. 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：

- FPGA 組使用 modelsim 進行模擬，在 compile verilog 時，使用下面指令：

```
vlog FAS.v  
vlog testfixture1.v 或 vlog testfixture2.v  
或  
vlog FFT_PE.v  
vlog testfixture0.v
```

- FPGA 組使用 modelsim 進行模擬，在 compile VHDL 時使用下面指令：

```
vcom FAS.vhd  
vlog testfixture1.vhd 或 vlog testfixture2.v  
或  
vcom FFT_PE.vhd  
vlog testfixture0.v
```

- 若 RTL 模擬時，為了確保足夠的可視範圍(visibility)，可在載入時加上「-novopt」參數以關閉最佳化功能，為避免時序檢查以減少錯誤訊息，可於模擬指令中加入 +notimingchecks，其參考指令語法如下(work 為工作 library 之名稱，test 為測試樣本的 module name):

```
vsim +notimingchecks -novopt work.test
```

- 提醒 VHDL 的使用者，如同參賽手冊所載明：比賽時使用 CIC 提供之 Mentor Graphics 公司的正式版 ModelSim。該版本之 ModelSim 有支援 Mix-language (即整個系統中混用 Verilog 及 VHDL 作設計)的模擬功能，為保持測試樣本之一致性，故未再提供 VHDL 版的測試樣本與 memory simulation model，但並無妨礙。參賽者只要掌握「對設計中的.v 檔用 vlog 作 compile」、「對設計中的.vhd 檔用 vcom 作 compile」、「最後在載入模擬時，不分 verilog 或 VHDL，一律用 vsim +notimingchecks -novopt work.tb 指令」的基本原則即可，細節請自行參考 ModelSim 使用手冊。若參賽者未詳閱參賽手冊之說明並遵行之，而使用自行下載之其它版本 ModelSim 軟體(如 Altera 或 Xilinx 公司提供的 OEM 版 ModelSim)，致使比賽時設計環境出問題造成不便，請自行負責。

附錄 C 快速傅利葉轉換公式推導（僅供參考，可略過）

FFT 原始數學表示式如公式二所示。

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

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

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

公式一

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二。

表八 DFT 與 FFT 運算元個數

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

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

公式二

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 $(-1)^k$ 如公式三。

$$w^{\frac{N}{2}k} = (-1)^k$$

公式三

將此因子代入公式二，可以表示為公式四。

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

公式四

將 k 拆為奇數部分與偶數部分，可得到公式五。

$$\begin{aligned}
 y(2r) &= \sum_{m=0}^{\frac{N}{2}-1} [x(m) + x(m + \frac{N}{2})] w^{m2r} \\
 y(2r+1) &= \sum_{m=0}^{\frac{N}{2}-1} [x(m) - x(m + \frac{N}{2})] w^m \bullet w^{m2r} \\
 r &= 0, 1, \dots, \frac{N}{2} - 1
 \end{aligned}
 \tag{公式五}$$

並可將公式六代換表示為公式六

$$\begin{aligned}
 x'0(m) &= x(m) + x(m + \frac{N}{2}) \\
 x'1(m) &= [x(m) - x(m + \frac{N}{2})] w^m
 \end{aligned}
 \tag{公式六}$$

最後，可推導出公式七，而在硬體實現上，架構如圖十六。

$$\begin{aligned}
 y(2r) &= \sum_{m=0}^{\frac{N}{2}-1} x'0(m) w^{m2r} \\
 y(2r+1) &= \sum_{m=0}^{\frac{N}{2}-1} x'1(m) w^{m2r}
 \end{aligned}
 \tag{公式七}$$

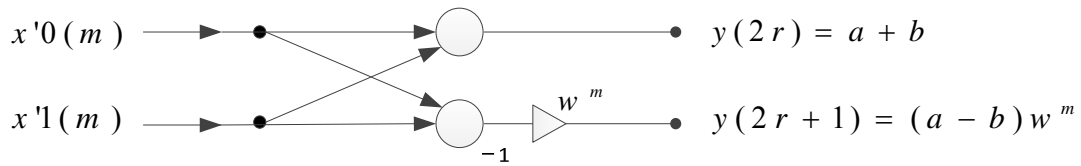


圖 十六、2 radix 蝶型快速傅利葉轉換硬體架構

其 2 點及 4 點快速傅利葉轉換，硬體架構如圖十七所示。

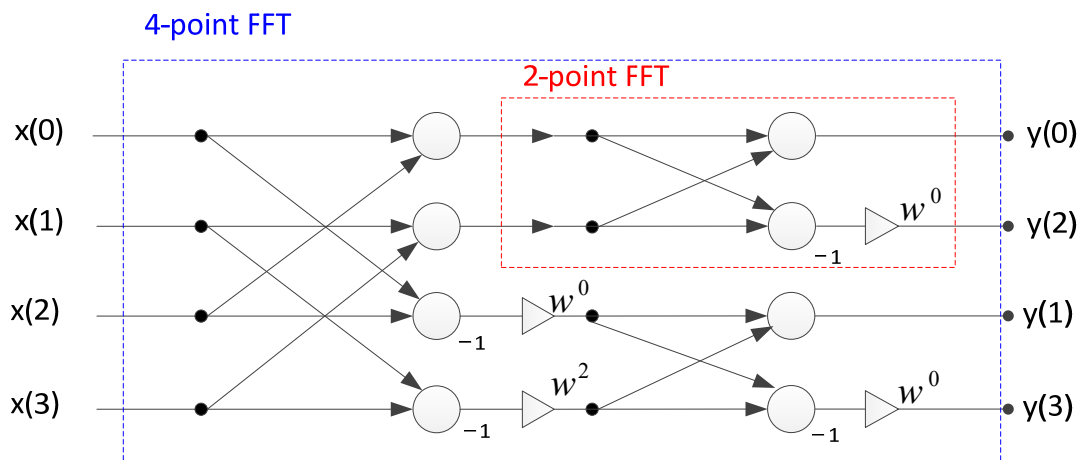


圖 十七、2 點及 4 點快速傅利葉轉換 (FFT) 硬體架構

其 8 點快速傅利葉轉換硬體架構如圖十八所示。

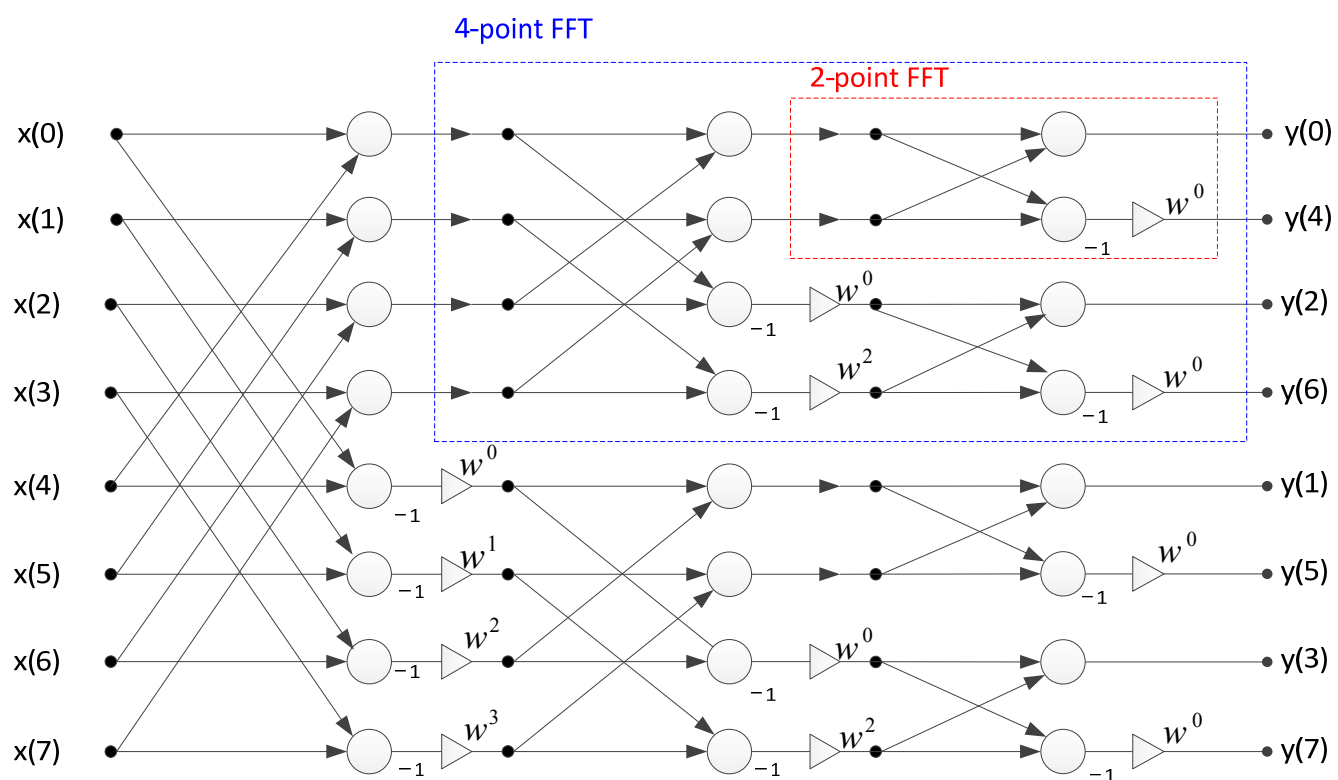


圖 十八、8 點快速傅利葉轉換 (FFT) 硬體架構

附錄 D 評分用檔案

如表八所示，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)Post-layout gate-level netlist，即由 FPGA 發展軟體所產生的 Post-layout gate-level netlist，以及對應的 SDF 檔(Altera 組為*.vo、*_v.sdo 檔或*.vho、*_vhd.sdo 檔，Xilinx 組為*_timesim.v、*_timesim.vhd 及 *_timesim.sdf 檔)；(3)report file，參賽隊伍必須依照自己的設計內容，撰寫 report_XXX.txt 檔，以方便主辦單位進行評分，report_XXX.txt 的格式如圖十二所示。(report 檔以後三碼序號表示版本，首次繳交為 report_001.txt。若繳交檔案更新版本，則新版的 report 檔檔名為 report_002.txt，依此類推)。**請注意**，本次初賽之電路設計並未開放使用 FPGA 廠商之 IP(如 Altera 之 Megafunction Wizard 及 Xilinx 之 Core Generator)，若參賽者之設計中有使用到 Altera 的 Megafunction/Xilinx 的 Co-gen & Architecture IP 的話，將不予計分。

表八、評分用檔案

<i>Result_XXX directory</i>	
<i>RTL category</i>	
<i>File format</i>	<i>Description</i>
*.v or *.vhd	Verilog (or VHDL) synthesizable RTL code
<i>Post-Layout Gate-Level category</i>	
<i>File format</i>	<i>Description</i>
*.vo, *.vho, *.sdo (*v, *.vhd, *.sdf)	Verilog/VHDL post-layout gate-level netlist generated by FPGA Development Tool, and SDF file
<i>Report category</i>	
<i>File format</i>	<i>Description</i>
report_XXX.txt	Design report file

舉例來說，假設參賽隊伍採 Verilog 設計語言，並以模組化設計將系統切分為 TOP.v、sub1.v 及 sub2.v，經 FPGA 發展軟體(以 Quartus II 為例)合成後所產生的 post-layout gate-level netlist 為 TOP.vo，所產生的 SDF 檔為 TOP.sdo，則參賽隊伍須將 TOP.v、sub1.v、sub2.v、TOP.vo、TOP_v.sdo、report_XXX.txt 等檔案確實放置於同一個 result_XXX 目錄下。其中 report_XXX.txt 的內容可參考下面圖十九所示：

FTP 帳號: 60001

--- RTL category---

使用之 HDL 語言:

Verilog

使用之合成器:

Quartus II

RTL 檔案名稱(RTL filename):

TOP.v, sub1.v, sub2.v

Post-layout gate-level Netlist:

TOP.vo, TOP_v.sdo

Report:

report_001.txt (此次為第一次上傳)

-----注意事項(annotation)-----

(其餘注意事項依各參賽隊伍的需求填寫，非為必要部份)

TOP 電路 RTL 設計檔之 Top module 為 TOP，位於 TOP.v 內

圖 十九、report.txt 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*.rar 格式(建議以*.zip 格式為優先)¹。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.txt”(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.txt”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.txt”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.txt 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*.zip(或*.rar)格式，如 Result_xxx.zip(或 Result_xxx.rar)。
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.txt 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!

¹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

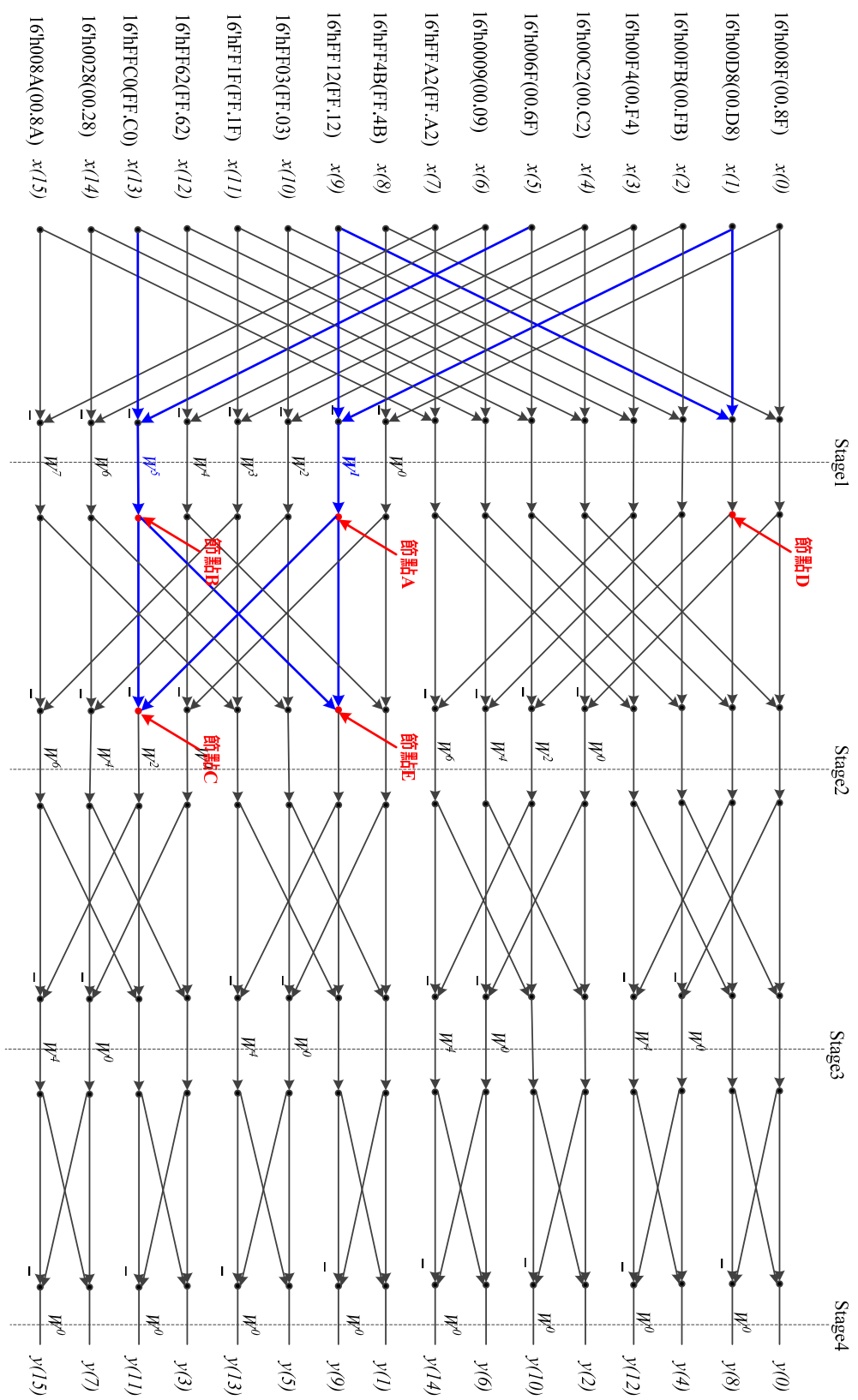


圖 二十、16 點 FFT 運算說明

節點 A=[x(1)-x(9)]*W¹

=[16'h00D8+16'h0j-(16'hFF12+16'h0j)]*(32'h0000EC83+32'hFFFF9E09j)

(a b c d W1_real W1_imag)

依推導結論

實數=(a-c)Wn_real+(d-b)Wn_imag

=(16'h00D8-16'hFF12)*32'h0000EC83+(16'h0-16'h0)*32'hFFFF9E09

=32'h01A37052 (01.A37052)

虛數=(a-c)Wn_imag+(b-d)Wn_real

=(16'h00D8-16'hFF12)*32'hFFFF9E09+(16'h0-16'h0)*32'h0000EC83

=32'hFF5243F6(FF.5243F6)

最後將實數與虛數資料整合，整數與小數各為 8bit

節點 A 輸出結果為 32'h01A3FF52 ({16 位元實數資料,16 位元虛數資料})

節點 B=[x(5)-x(13)]*W5

=[16'h006F+16'h0j-(16'hFFC0+16'h0j)]*(32'hFFFF9E09+32'hFFFF137Dj)

(a b c d W5_real W5_imag)

依推導結論

實數=(a-c)Wn_real+(d-b)Wn_imag

=(16'h006F-16'hFFC0)*32'hFFFF9E09+(16'h0-16'h0)*32'hFFFF137D

=32'hFFBD0827 (FF.BD0827)

虛數=(a-c)Wn_imag+(b-d)Wn_real

=(16'h006F-16'hFFC0)*32'hFFFF137D+(16'h0-16'h0)*32'hFFFF9E09

=32'hFF5E5273(FF.5E5273)

最後將實數與虛數資料整合，整數與小數各為 8bit

節點 B 輸出結果為 32'hFFBDFF5E ({16 位元實數資料,16 位元虛數資料})

節點 C=[節點 A-節點 B]*W2

=[16'h01A3+16'hFF52j-(16'hFFBD+16'hFF5Ej)]*(32'h0000B504+32'hFFFF4AFC)

(a b c d W5_real W5_imag)

依推導結論

實數=(a-c)Wn_real+(d-b)Wn_imag

=(16'h01A3-16'hFFBD)*32'h0000B504+(16'hFF5E-16'hFF52)*32'hFFFF4AFC

=32'h014F2968 (01.4F2968)

虛數=(a-c)Wn_imag+(b-d)Wn_real

=(16'h01A3-16'hFFBD)*32'hFFFF4AFC+(16'hFF52-16'hFF5E)*32'h0000B504

$$=32'hFE9FDE38(FE.9FDE38)$$

最後將實數與虛數資料整合，整數與小數各為 8bit

節點 C 輸出結果為 32'h014FFE9F ({16 位元實數資料,16 位元虛數資料})

節點 D= $x(1)+x(9)$

$$=(16'h00D8+16'h0j)+(16'hFF12+16'h0j)$$

(a b c d)

實數與虛數各別運算

$$\text{實數}=a+c$$

$$=16'h00D8+16'hFF12$$

$$=16'hFFEA$$

$$\text{虛數}=16'h0+16'h0$$

$$=16'h0000$$

最後將實數與虛數資料整合，整數與小數各為 8bit

節點 D 輸出結果為 32'hFFEA0000 ({16 位元實數資料,16 位元虛數資料})

節點 E=節點 A+節點 B

$$=(16'h01A3+16'hFF52j)+(16'hFFBD+16'hFF5Ej)$$

(a b c d)

實數與虛數各別運算

$$\text{實數}=a+c$$

$$=16'h01A3+16'hFFBD$$

$$=16'h0160$$

$$\text{虛數}=16'hFF52+16'hFF5E$$

$$=16'hFEB0$$

最後將實數與虛數資料整合，整數與小數各為 8bit

節點 E 輸出結果為 32'h0160FEB0 ({16 位元實數資料,16 位元虛數資料})

如表九，輸入前 16 筆資料進行 FFT 運算過程之結果。表九資料是由 Matlab 產生，故精確度較高。評分時所使用的測試樣本是允許整數與小數各 4bit 誤差。

表九、FFT 運算結果

	Stage1		Stage2		Stage3	
	實數	虛數	實數	虛數	實數	虛數
0	16'hFFDA	16'h0000	16'hFFFE	16'h0000	16'h002D	16'h0000
1	16'hFFEA	16'h0000	16'h0019	16'h0000	16'h0058	16'h0000
2	16'hFFFE	16'h0000	16'h002F	16'h0000	16'hFFCF	16'h0000
3	16'h0013	16'h0000	16'h003F	16'h0000	16'h0000	16'h0026
4	16'h0024	16'h0000	16'hFFB6	16'h0000	16'hFFB6	16'h0033
5	16'h002F	16'h0000	16'hFFCF	16'h0030	16'hFFE0	16'h0042
6	16'h0031	16'h0000	16'h0000	16'h0033	16'hFFB6	16'hFFCD
7	16'h002C	16'h0000	16'h0011	16'h0011	16'h001F	16'h0042
8	16'h0144	16'h0000	16'h0144	16'hFEA0	16'h02BE	16'hFD51
9	16'h01A3	16'hFF52	16'h0160	16'hFEB0	16'h02EA	16'hFD58
10	16'h0164	16'hFE9B	16'h017A	16'hFEB1	16'hFFC9	16'hFFEE
11	16'h00B3	16'hFE4E	16'h0189	16'hFEA7	16'h0009	16'h0029
12	16'h0000	16'hFEA0	16'h0144	16'h0160	16'hFFC9	16'h0011
13	16'hFFBD	16'hFF5E	16'h014F	16'hFE9F	16'hFFF6	16'h0029
14	16'h0015	16'h0015	16'hFE85	16'hFEB1	16'h02BE	16'h02AE
15	16'h00D6	16'h0058	16'hFEA7	16'h0189	16'hFD15	16'hFD58

	FFT Data Out	
	實數	虛數
fft_d0	16'h0085	16'h0000
fft_d1	16'h05A8	16'hFAAA
fft_d2	16'hFF97	16'h0075
fft_d3	16'hFFC1	16'h003A
fft_d4	16'hFFCF	16'h0026
fft_d5	16'hFFD3	16'h0017
fft_d6	16'hFFD6	16'h000F
fft_d7	16'hFFD5	16'h0006
fft_d8	16'hFFD5	16'h0000
fft_d9	16'hFFD5	16'hFFFA
fft_d10	16'hFFD6	16'hFFF1
fft_d11	16'hFFD3	16'hFFE9
fft_d12	16'hFFCF	16'hFFDA
fft_d13	16'hFFC1	16'hFFC6
fft_d14	16'hFF97	16'hFF8B

fft_d15	16'h0085	16'h0556
---------	----------	----------