**數位電路實驗**

**Final Project – Draw Your Life**

**第七組：B04901068劉力仁**

**B04901111洪鈺萌**

**B04901013李旻芳**

##### **目錄**

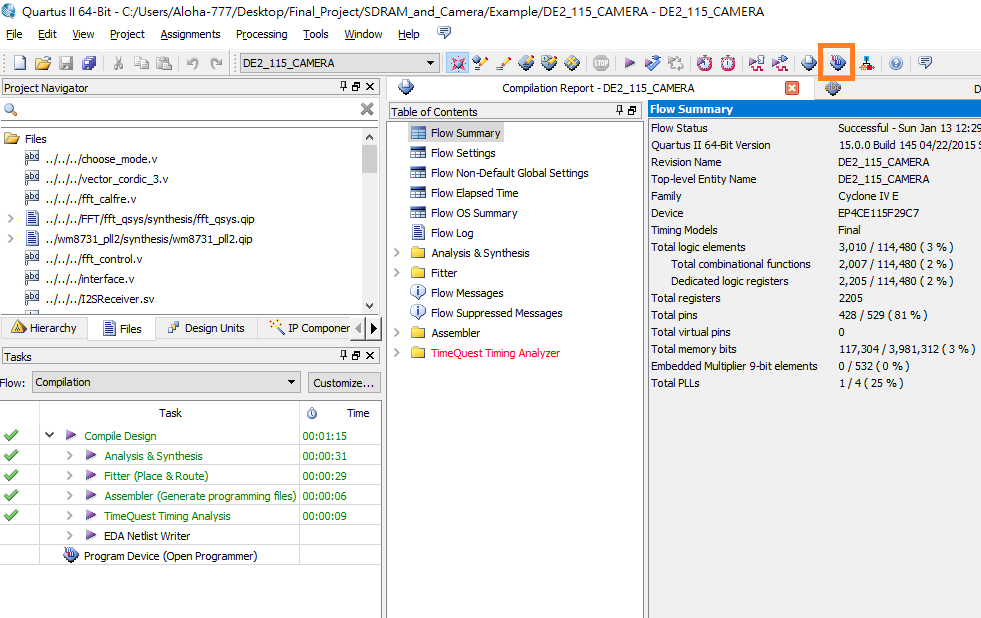
1. **使用說明**
2. **提要**
3. **主要功能**
4. **使用器材**
5. **操作說明**
6. **期末專題介紹**
7. **動機**
8. **原理**
9. **遇到的問題與解決**
10. **未來展望**

##### **壹、 使用說明**

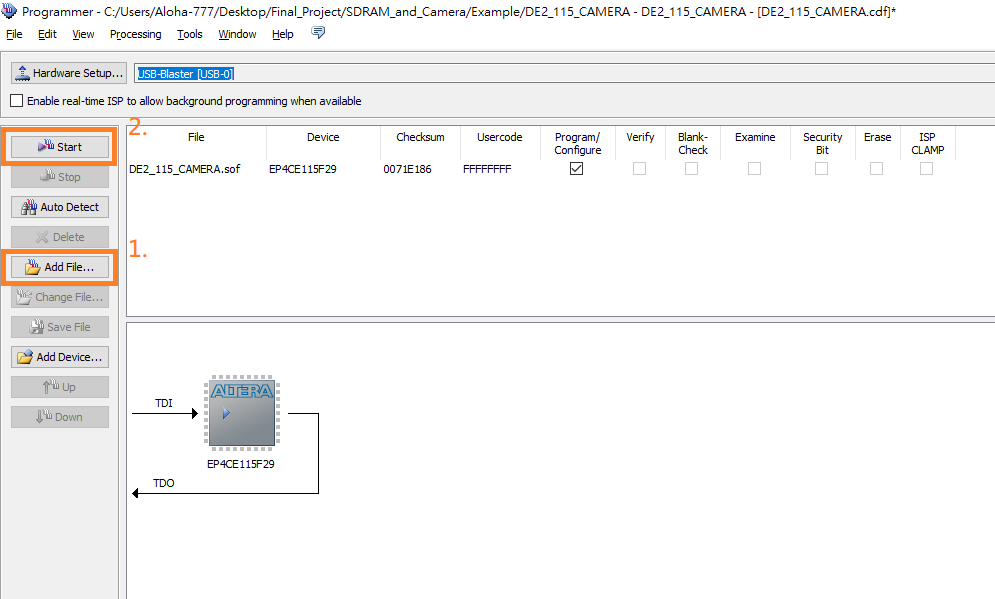
1. **提要**

**本專題主題為Draw Your Life，目的是以FPGA基本的影像功能，搭配D5M照相機，佐以自行設計的繪圖介面，包含選擇圖形、顏色、橡皮擦等等功能，達到使用者能以real-time且幀率為15 fps的速度，直接於VGA投影出之D5M相機拍攝畫面上繪畫。**

1. **主要功能**
2. **透過D5M照相機捕捉當前畫面上色點的位置，讓使用者能以色點作為繪畫工具隔空作畫。**
3. **照相機擷取到的當下畫面會以VGA同步顯示在螢幕上。**
4. **螢幕左側顯示可選擇之繪圖工具，方便使用者做選取。**
5. **繪圖工具包含矩形、三角形、直線模式，搭配紅、橙、黃、綠等等基本色彩，以及可選取透明色以當前模式作為橡皮擦功能或是直接清除全畫面。**
6. **聲音輔助系統，附有麥克風，透過唱出不同的聲調，快速選擇當前畫筆的顏色。**
7. **使用器材**
8. **DE2-115板子（FPGA）**
9. **VGA螢幕及傳輸線**
10. **D5M照相機**
11. **包含至少紅光、綠光之光源**
12. **操作說明**
13. **燒錄檔案至FPGA**
14. **首先需下載Quartus II，下載完畢後進入Quartus，接著點選下圖橘色框處按鈕，進入燒錄頁面。**

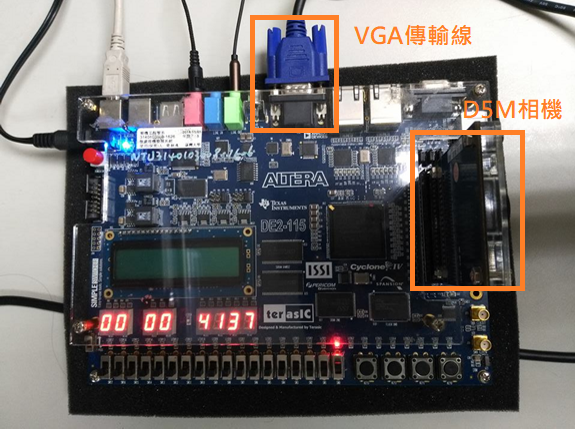
****

1. **先確保FPGA已開啟電源且連接USB傳輸線至電腦，接著於燒錄頁面點選Add File…，加入DE2\_115\_CAMERA.sof，並按下Start鍵，即完成燒錄動作。**

****

1. **相機與VGA**

**待FPGA燒錄完成後，將D5M相機接於FPGA特定腳位上，並透過VGA傳輸線將影像資料接到欲顯示的螢幕上，此時螢幕上所顯示之畫面即為當前D5M相機所讀取到的影像，並且於畫面左方額外顯示使用者繪圖介面。**

****

1. **按鍵功能**

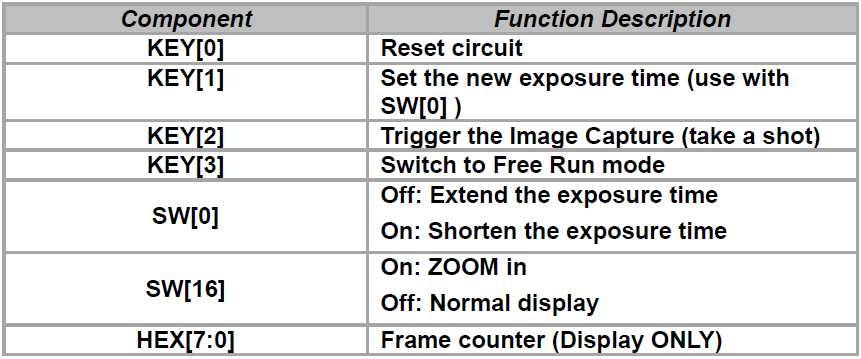
**下表為相關按鍵之功能（註1）：**

**KEY[0] 可初始化界面，回到最一開始狀態；**

**KEY[1] 搭配SW[1] 可調整畫面亮度；**

**KEY[2] 畫面暫停、KEY[3] 畫面開始；**

**SW[16] 開關調整完後需再按下KEY[0] 才能改變畫面遠近。**

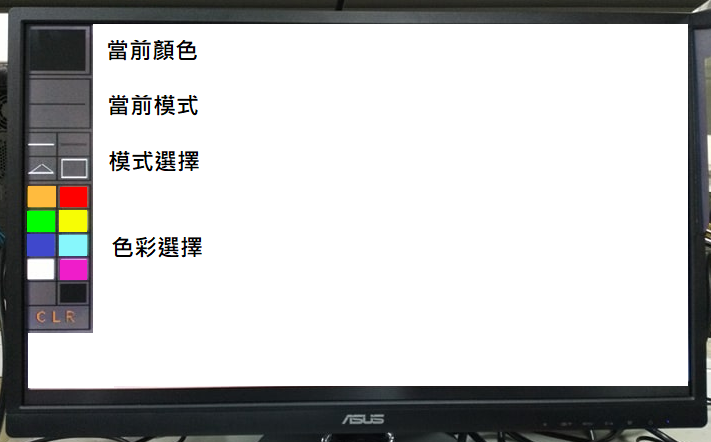
****[[1]](#footnote-1)

1. **繪圖功能**
2. **偵測紅綠光源**

**將預先準備的紅光與綠光光源對準相機，此時螢幕上會顯示光源位置。若相機偵測到綠光可於繪圖介面選取功能；而偵測到紅光則可於當下繪圖狀態在背景上顯示圖形。注意若相機無法偵測光源，可先以KEY[1] 與SW[1] 按鍵將螢幕亮度調暗，可提高偵測的靈敏性。**

1. **繪圖介面**

**螢幕左方繪圖介面由上至下依序為：當前顏色、當前模式、模式選擇、色彩選擇。而模式選擇由右至左、由上至下依序為：直接模式（偵測到任意紅光位置即顯示色點在螢幕上）、直線模式（將偵測到之任2點自動以直線連接）、矩形模式（偵測2點位置會顯示該2點所形成之矩形圖形，而2點間的時間間距約為6秒左右）、三角形模式（偵測2個紅光位置，形成三角形，2點間隔時間亦為6秒）。而顏色選擇特別的是左下角(黑色旁邊)的透明顏色即為橡皮擦，會搭配當前使用者所選的模式執行橡皮擦的功能，可把先前所畫的圖形擦拭掉。至於選取最下方最大格的透明格(CLR)可清除當前繪圖畫面。**

****

1. **聲音輔助功能**

**麥克風收取使用者音訊訊號，當頻率波段在150 Hz至300 Hz時，能改變繪圖介面的「當前顏色」，由低至高分別顯示紅、橙、黃、綠、藍、靛、紫、白、黑、透明。音訊區段約為15 Hz，維持一定區間內的音頻0.5秒即會選擇顏色。**

##### **貳、期末專題介紹**

1. **動機**

**之所以會做類似小畫家的繪圖軟體，其實我們考量許多，原因如下：**

1. **喜歡美的事物**

**普遍大眾喜愛美的東西，這是無可否定的。我們這組希望能實現即時的繪圖工具，讓使用者可以隨時為自己的生活增添更多的色彩。**

1. **增進小畫家功能**

**一般小畫家給人的印象是一張白紙，可以在紙上繪製各樣的色彩。但我們希望，繪圖的區域不再只局限於一張白紙，而可以擴充到生活的每一個片段。**

1. **熟悉更多FPGA的基本應用**

**實驗一實做簡單點名器，我們熟悉了基本的按鈕與七段顯示器的使用；實驗二做的是複雜的加、解碼器，我們熟悉複雜的演算法與RS-232之間的傳輸；實驗三則做簡單錄放音器，我們熟悉了SRAM與音訊晶片的使用。然而，三個實驗缺少了基本的影像處理與應用，我們希望能在期末專題熟悉這部分的使用。此外，我們也發現要完成一個project往往需要使用別人做好的IP core，以避免自己重新發明一個輪子（re-inventing the wheel），因而決定嘗試使用Qsys提供的IP core。在音訊處理的部分，我們使用Qsys的Fast Fourier Transform IP core，學習看User Guide並學習用ModelSim進行驗證，讓我們在做project的過程中學會運用更多工具。**

1. **先前數電實驗相關題目之組別的啟發**

**在決定我們的題目前，我們參考了歷年實驗的成果，其中 [2014 Spring] Laser Printer算是與我們最類似的成果，然而，當時他們的繪圖區域只侷限在一個小箱子內，並且他們不需考慮實際背景而是僅僅於白紙上作畫，相對訊號的處理就簡單了許多，我們希望能以他們的成果為基礎，額外增加更多巧思。**

1. **實際應用**

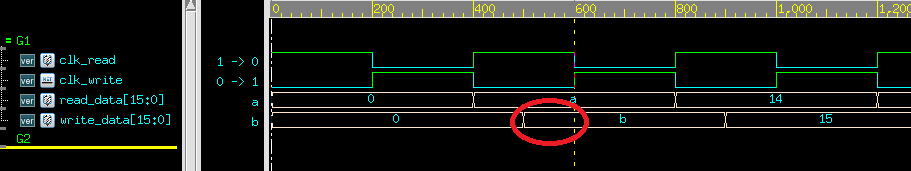
**能即時於當前背景作畫的功能，若可解決畫素低以及操作不太方便的問題，即可實現遠距離視訊的互動功能，讓其中一方得以於鏡頭前方數公尺的距離，即於螢幕上畫上要傳達給對方的圖案與資訊。**

**而之所以選用光源和音訊作為我們的繪圖工具，是因為相較於觸控筆，這兩種方式能夠在遠距離使用，而不用在接受器（相機、觸控板）前，雖然精確度不會像手機的觸控功能，這是可以拿來用在較大距離的應用，像是教學、實況主播，算是性質不太相同的作品。**

1. **原理**
2. **SDRAM讀取與VGA顯示**

**跟先前大家做過的project不同點在於，我們需要同時儲存相機的照片和畫筆所畫的圖片，共兩張800\*600像素的圖片。一張圖片的資訊量太大，不可能使用暫存器來存，因此我們將兩張圖片都存進SDRAM裡。原本以為這是蠻簡單的事，但實作後發現不然，因為原本的SDRAM控制很複雜，它同時要接收相機傳來的圖片，還要傳給VGA顯示，裡頭用到FIFO等機制，我們無法更動。因此我們很難另外找一個記憶體存畫筆畫的圖片，並使它和相機所照的圖片同步輸出。我們最後採用捨棄精確度的方法，將原本存入32bits顏色的相機圖片，變成16bits的相機圖片和16bits的畫筆圖片（如果FPGA的板子SDRAM頻寬是64bits而不是32bits我們就不用捨棄）。**

**其中，寫入畫筆的圖片的clk改成SDRAM讀取端clk的反向，也就是晚半個週期，如此一來，才可以使SDRAM把讀出的畫畫圖片，在同一時間存入對應位置的pixel（因為寫入和讀出都會經由FIFO，沒辦法自由控制地址，因此採用這個折衷辦法）下圖為寫入clk和讀取clk的波形圖：**

****

**可以發現從SDRAM讀出的值需要在半個周期內運算完，才能在寫入clk升起之前，將答案傳給SDRAM做寫入的工作。**

**SDRAM讀取與VGA顯示的方法基本上和NAS提供的相機模組project差不多，只是要注意幾點：**

**第一點：要偵測照相機抓到的色點，即時存入。**

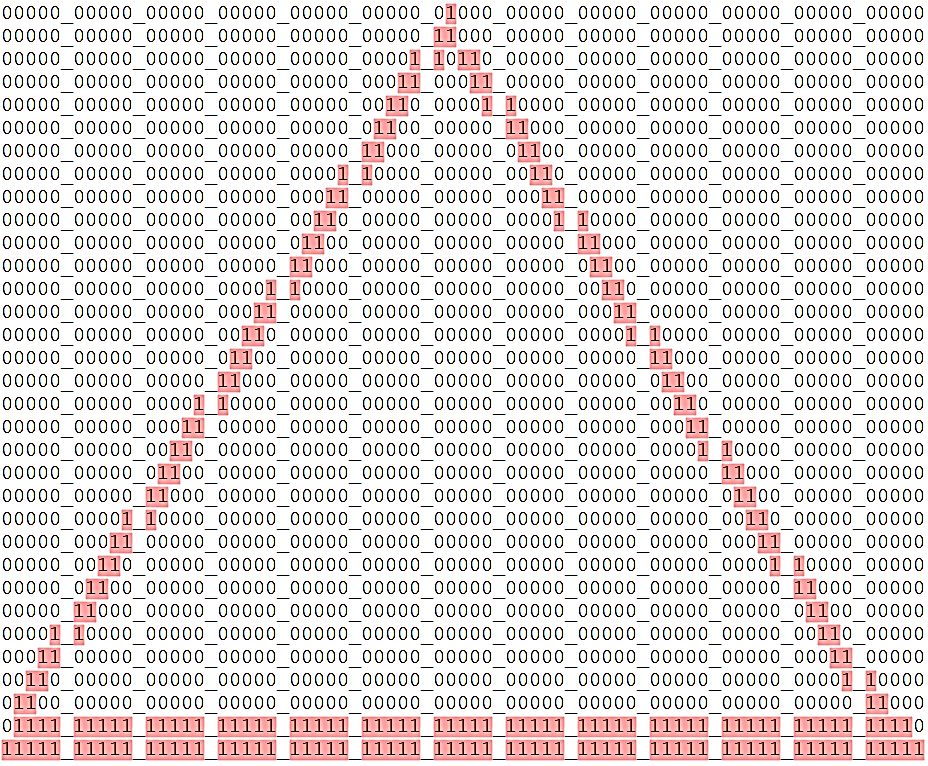
**第二點：要在相同的時間抓出相機拍到的圖片和畫筆所畫的圖片，做運算後傳給VGA  
 顯示。**

**第三點：畫筆所畫的圖案要能夠維持，因此讀出後必須和畫畫的module溝通，決  
 定要存既有的值（原本的畫），或是新畫的線條，在半個周期內。**

1. **介面顯示**

**於螢幕左方顯示繪圖介面的實現，其實原理滿簡單的，複雜之處在於需對VGA顯示的座標系統有一定的了解，才能透過事先定好的X、Y座標，讓VGA掃到該點時能顯示對應的顏色，以確保圖形是在自己所期望的位置出現。另外，以實心圖形顯示也比以空心圖形顯示來得簡單，但為了追求整個介面能更有質感，在顯示圖形模式上我們以空心顯示，但判斷式的撰寫上也因此變得更加複雜，畢竟要考慮透明處為VGA不上色的區域，而非透明處則需以當前狀態讓VGA顯示相對應的色彩。**

**而所有在介面顯示的線段，我們皆加寬為2個pixels點的寬度去做顯示。另外，圖形中又以顯示空心三角形最為複雜，畢竟其2邊為有斜度的直線，又需確保線寬看起來有2個pixels的寬度，而這部份我們採取的是先定好三角形的邊界，再以python跑出該範圍內需顯示的點，將所有欲顯示的點寫成一個資料庫後（如下圖，數字1表示需要顯示之點；數字0則表示不需顯示），即可以配對的方式決定該點是否顯示顏色，大幅縮減了程式的複雜性，以類似的方法便可畫出各種更複雜的圖形。**

****

1. **偵測光點與游標**

**當D5M相機擷取到當前畫面，首先會將畫面資訊儲存到SDRAM，而偵測光點的部分是取SDRAM所儲存的影像資料為判斷依據，我們會先設定一個固定的色彩閾值，假如SDRAM所傳來的某一色點其顏色RGB值落於該範圍內（表示使用者開啟某顏色光源），即會被偵測到。並且為了方便，我們設定每一個frame的影像只允許有一個光點被偵測到，因此只會取第一個被偵測到的光點，其餘的光點則捨去，而所偵測到的光點則作為使用者畫圖與選取工具的依據。**

**至於游標則是偵測到光點後的擴充顯示，我們會將所偵測到的光點位置往右與往下擴充成3 pixels \* 3 pixels大小的正方形，並且設定游標顯示優先於其他色彩之顯示，即可協助使用者透過觀察游標位置來畫圖與選取工具。**

**另外需注意的是，假如畫圖與選功能是以同一閾值的色點作為判斷，會導致使用者可能只是想選功能，卻不斷於螢幕上繪圖的問題發生，因此畫圖與選功能我們設定了2個不同的閾值範圍，這也是為什麼本專題需使用至少包含紅光（畫圖）與綠光（選功能）2種光源的原因。**

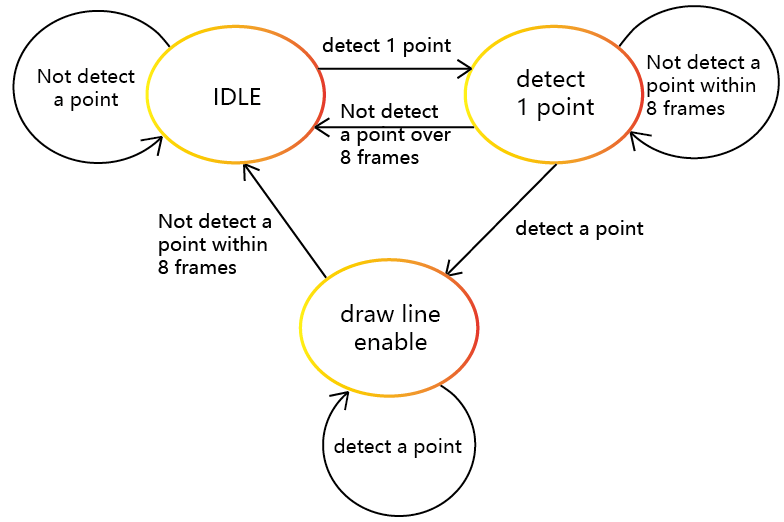
1. **圖形功能**
2. **直接模式**

**此模式基本上與偵測光點的處理方式幾乎相同，當有紅光光點被偵測到，即在螢幕上對應的位置顯示當下所選顏色，而此模式所繪製出的圖形有很大部分與所使用的光源有關，會隨著光源而改變形狀。**

1. **直線模式**

**此模式的處理方法是當2紅光光源點被偵測到，即透過2點之間的斜率去計算需自動補上哪些點，而一般而言，由於斜率大小並非整數，但螢幕的X、Y座標皆是整數，為了方便處理，所有運算我們皆只考慮整數，捨去小數的部分。另外，直線的寬度我們設定為3個pixels的大小，由於需同時考慮線寬、斜率大小、斜率正負、2個偵測點的順序等等，使得在處理直線上變得相當麻煩，有很多不同的情況需要考慮。**

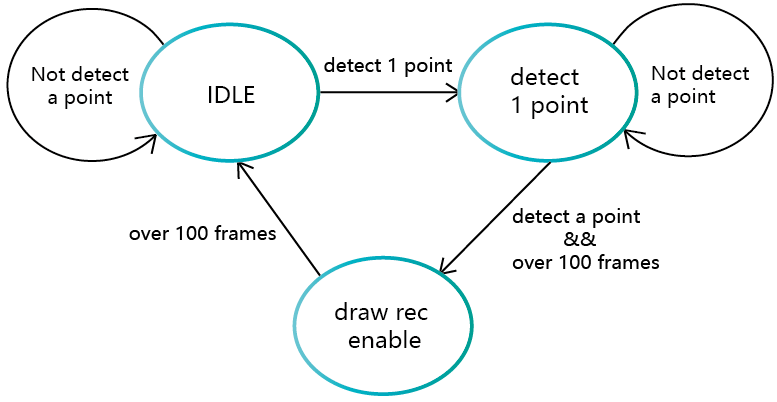
**而控制直線模式的finite state machine如下圖，起初會在IDLE模式，當偵測到1個色點會進到detect 1 point，此時若在8個frames內偵測到另一個色點，即可開始畫線。而需注意若在任何模式下連續超過8個frames皆無偵測到色點，會強迫進到IDLE模式，表示使用者可能開始畫不同位置的直線或是停止畫線。**

****

1. **矩形模式**

**畫矩形比畫直線來得簡單許多，一樣是透過偵測2個點，將2點所圍成的矩形補出來，且寬度亦為3個pixels，但由於矩形的四邊皆垂直，因此在判斷該補哪些點上較為容易，沒有不同斜率的問題。**

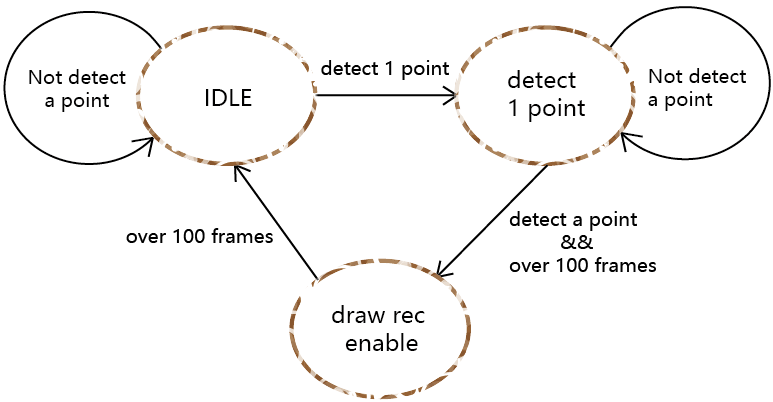
**控制矩形的finite state machine如下，一開始一樣處於IDLE state，當偵測到1點後進入detect 1 point，而下一點偵測需等100個frames以後（約莫6秒），以確保使用者有足夠的時間決定矩形位置，另外，每畫完一個矩形仍需等100個frames後才能再畫下一個矩形。**

****

1. **三角形模式**

**三角形模式與矩形模式大致相似，寬度亦為3個pixels，只不過其尚需考慮斜率問題，且2點的偵測上第一個點固定為頂點，與第二個點決定側邊的斜率，會根據鉛直軸補上另一邊，再加上底邊而形成等腰三角形，由於亦有斜率問題，因此處理上也較為麻煩。**

**而控制三角形的FSM如下圖，與矩形的大致相同，在偵測第二個點與下一個三角形上亦需等待100 frames才可進行偵測。**

****

1. **顏色選取**

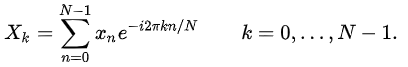
**以上繪圖功能可搭配不同顏色作顯現，且使用者額外畫上的圖案會先於相機擷取的圖案顯示在VGA上，但若選取透明色，則轉而使畫過的地方為相機擷取的圖案優先顯示，即達成橡皮擦功能。同理，若選取最下方的CLR模式，則可將全部繪圖畫面以相同方法清除。**

1. **音訊處理**

**我們會加入音訊控制，是因為聲音和光線一樣，是互動式介面很常見的控制方法，且適合在中距離使用（比起按按鈕或觸控筆遠，但較無線電等近）。我們原本想做聲音辨識，然而看前幾年的作品，發現訓練聲音辨識的過程，大部分還是得用電腦訓練，且正確率不是很好（四種不同的音訊，正確率80%）。所以我們後來選用音頻控制，也就是用FFT將時域的音訊訊號轉換成頻域，來控制介面，達到穩定十種以上的頻率分辨。**

1. **傅立葉轉換**

**離散的傅立葉轉換公式如下：**

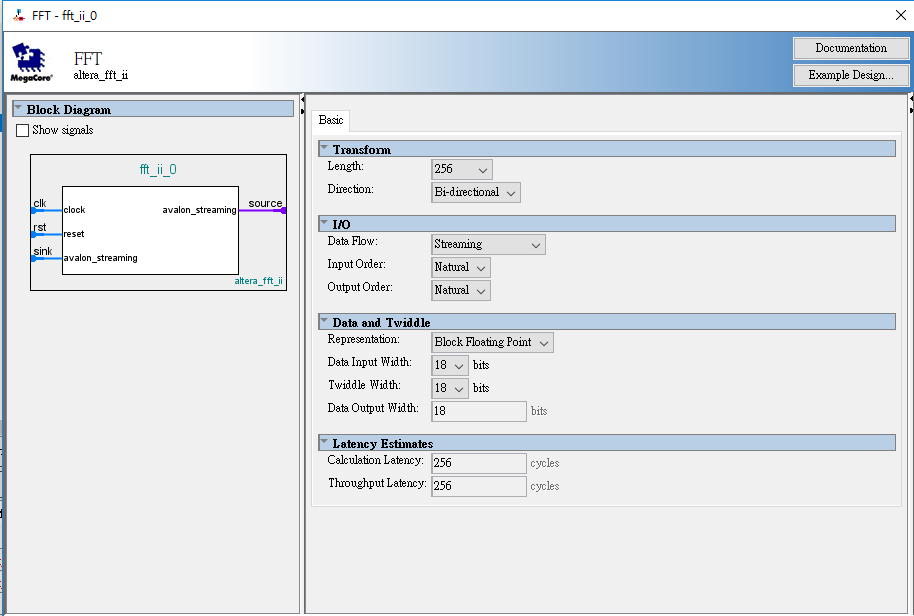
****

**為時域訊號，為頻域轉換結果，為取樣點數量。**

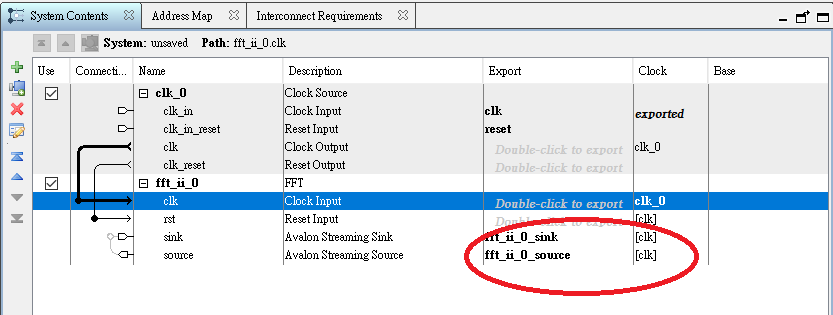
**如果時域中的取樣頻率為取樣點為，則傅立葉轉換後的基頻則是 ，根據取樣定理傅立葉轉換的解析度不會超過取樣頻率的一半，也就是。而~ 所對應的就是頻率為 的能量。**

1. **Qsys使用**

**實現方式是用Qsys合成的IP core，為256點的streaming 模式的FFT core。設定參數如下：**

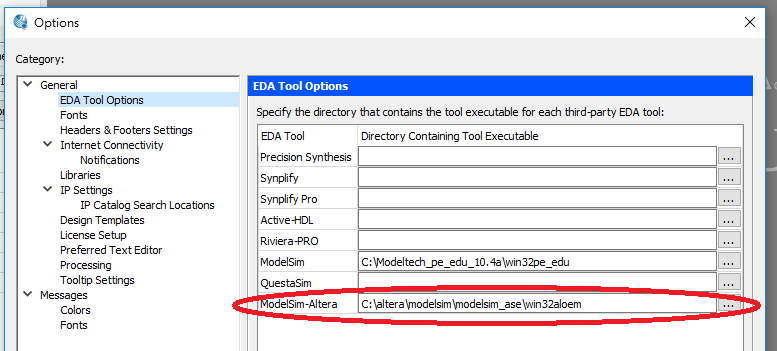
****

**接著把腳位外接**

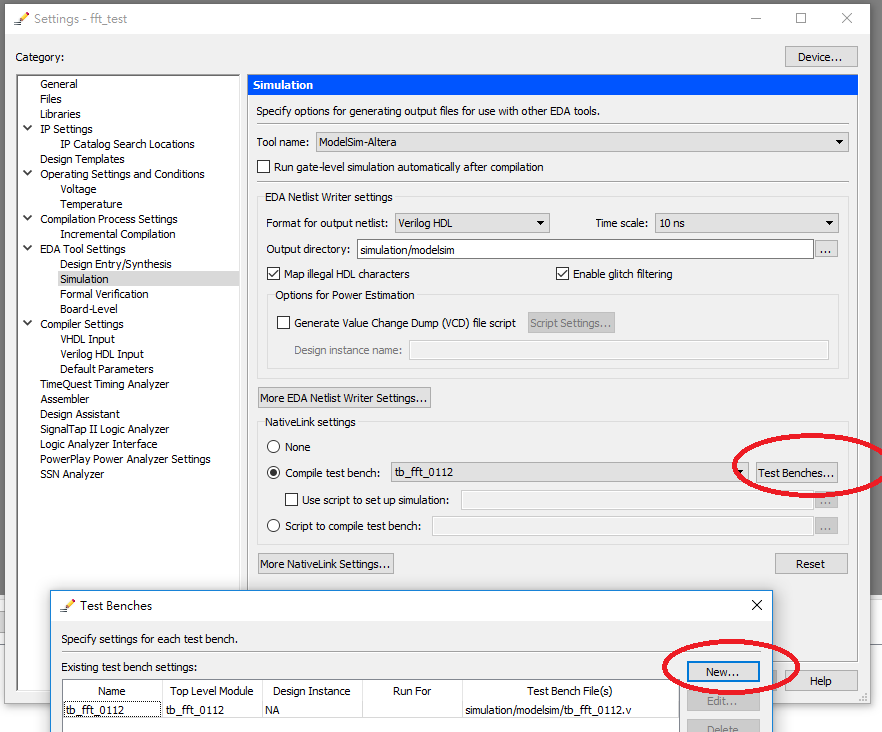
****

**使用方式可參考Quartus FFT User Guide**[[2]](#footnote-2)**，除錯方式可用Altera-ModelSim，配合NativeLink，可查閱官網。**

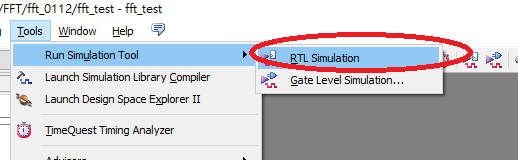
**要在Options中設定EDA tool路徑**

****

**並在setting中設定testbench**

****

**最後，在Tools中start RTL simulation**

****

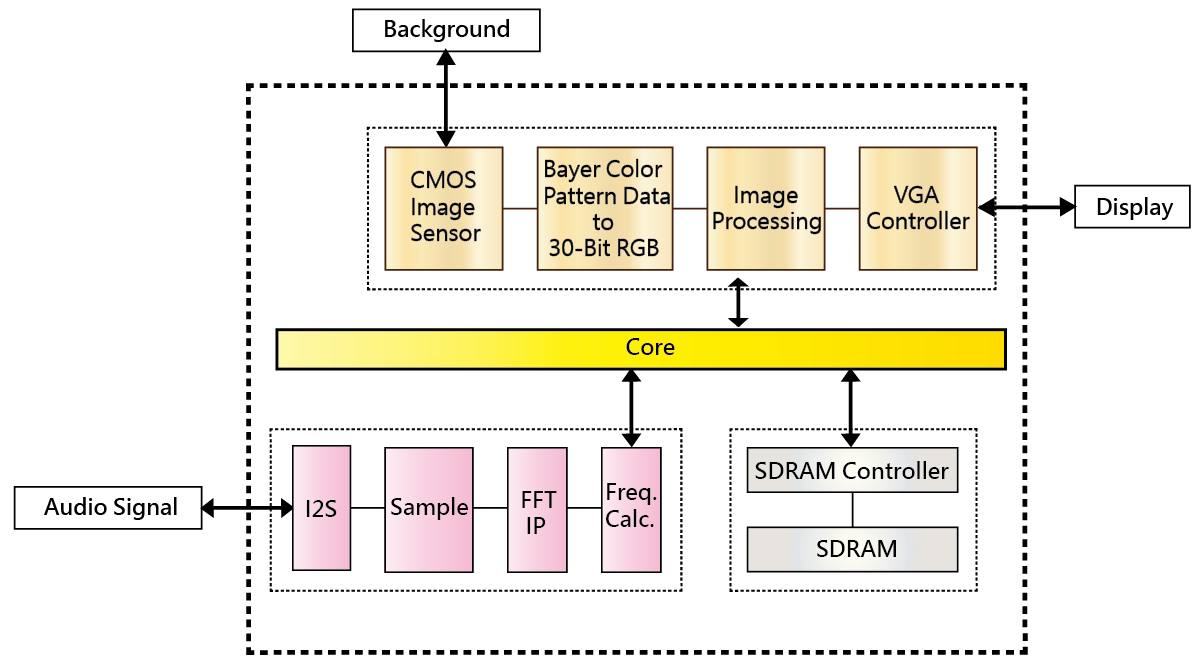
**之後就是使用ModelSim。以上是最簡單的驗證方法，其他用第三方模擬軟體都要設定更多東西。**

**合成成功後，用lab3所寫的I2S和I2C控制音訊晶片（wm8731），將收入的音訊訊號轉成16bits有號數，傳給FFT core做運算。**

1. **收取人聲頻率**

**人類雖然可以聽到20000 Hz的聲音，但實際上可發出的聲音約在** 80 ~ 1000 Hz。這跟麥克風750kHz的採樣頻率比起來，太低頻了，要採樣超過一萬點才有辦法精準偵測到人聲，而Qsys的FFT最多只能轉換1024 point的聲音，所以要降低採樣頻率。但也要注意如果是選用streaming模式的FFT core，輸入訊號是不能間斷的，否則會出錯。

1. **整體架構圖**

****

1. **遇到的問題與解決**
2. **色點偵測不易**

**原本在偵測色點上，我們預期可以透過使用者拿著一個特別顏色的物體或是直接於手指頭上貼上彩色膠帶作為畫筆，但後來發現這種方式很容易受到光線不足的影響，一旦角度稍微向下，即會導致亮度太暗而無法偵測到預期色點的RGB值，後來我們想既然光線不足，那不如使用特殊顏色的光源，如雷射筆、手電筒等等，即可自行提供穩定光源，解決這樣的問題。**

1. **讀寫時序問題**

**在SDRAM control中，寫入和讀取的地址是固定且一樣的我們沒辦法更改，SDRAM是不斷被新的輸入刷新的，如果要將畫筆畫過的畫維持住不被洗掉，寫入的值就必須和讀取的值相同，在沒辦法改地址的情況下我們只能更動clk，讓它們差半個週期，這樣寫入時才有辦法剛好抓取該pixel所存的顏色。**

1. **畫質太低**

**因為SDRAM頻寬只有32 bits，為了將每個frame偵測到的相機照片和畫筆圖片都存起來，必須將頻寬拆成兩半，結果色彩變得不鮮艷，顏色也不夠連續。我們嘗試過控制SDRAM的FIFO但太難了，裡面大部分是Qsys合成的code沒有邏輯，要用另一個記憶體來同步更是難上加難，所以最後只好用畫質不是很好的方法來實現我們的project。**

1. **螢幕易出現奇怪閃爍色點**

**當我們改動相機模組的project後，螢幕會發生各種奇怪的事，有時是有快速移動的黑點，有時是有類似邊界偵測的螢光，有時是螢幕整個變花花的。我們找到了兩個主要原因，一個是寫入SDRAM的時候因為讀取和寫入只差半個週期，如果中間放太多邏輯會造成時序問題，所以要盡量避免或擋暫存器。第二個原因是接腳位的時候，會把兩個位元數不同的接在一起，一開始我們以為編譯器會自己幫我們補，結果竟然會造成類似邊界偵測的螢光，讓我們除了很久的錯。**

1. **IP core的驗證**

**我們使用Qsys的FFT IP core最大問題是不知道怎麼驗證，雖然看了User Guide但不確定自己的控制訊號有沒有問題，將算出的頻率顯示在七段顯示器，結果很雜亂卻不知道是自己控制寫錯還是雜訊太多。我們後來用Altera提供的Modelsim來進行IP core的rtl驗證（註: Modelsim的官網只能下載學生版，如果是在Qsys上驗證，到Altera的官網找ModelSim-Altera來用比較簡單）。還有，傅立葉轉換的結果是以block floating point顯示，如果和自己用python或matlab結果跑出來不一樣也需要注意。**

1. **未來展望**
2. **頻率辨識轉聲音辨識**

**現在的聲音系統是用頻率做辨識，來對介面下指令，然而這樣並不直觀，且範圍有限，如果能將電腦上訓練好的聲音辨識模型用在FPGA上，也許能達到更佳的互動效果。**

1. **與網路結合**

**將目前的介面控制連上網路，讓我們除了畫圖外，還可以和他人交流。且在這個物聯網的時代，希望能用這個介面輕鬆的控制周遭電器、通訊設備。**

1. **TRDB-D5M User Manual:**

   <https://www.terasic.com.tw/attachment/archive/281/TRDB_D5M_UserGuide.pdf> [↑](#footnote-ref-1)
2. h**ttps://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/ug/ug\_fft.pdf** [↑](#footnote-ref-2)