FPGA系統設計

作業一

班級:資訊工程學系博士班一年級

組成員:104582004王建鈞

班級:資訊工程學系碩士班一年級

組成員:104522040張瑞慶、104522065張翔珳

實驗目的: 設計一系統輸入角度值，透過泰勒級數展開計算COS(角度值)並輸出。

方法一 IEEE 754

1. 實驗方法: 使用了IEEE 754 Library所以會有float型態，依照題目所需input為(3 downto -2)，output為(3downto -6)，其中負數為小數部分，整數為正負號與指數部分。

# ModelSim Simulation & Matlab



紅色點為系統輸出點，藍色線為Matlab cos圖形，可以看出連續好幾筆輸入會被視為同一輸入值，因為3to-2只有6bits能夠表示的太少了，導致使用IEEE754會造成小數值誤差不大時，從real轉成float會把差異不大的值轉成同樣的float值，所以輸出結果相同，但若取樣來看，整體是符合藍色線的。



從上圖來看，每個S0產生一次方波代表有一次輸入，輸入值為itheta，而系統有連續好幾個輸入皆為一樣的值。

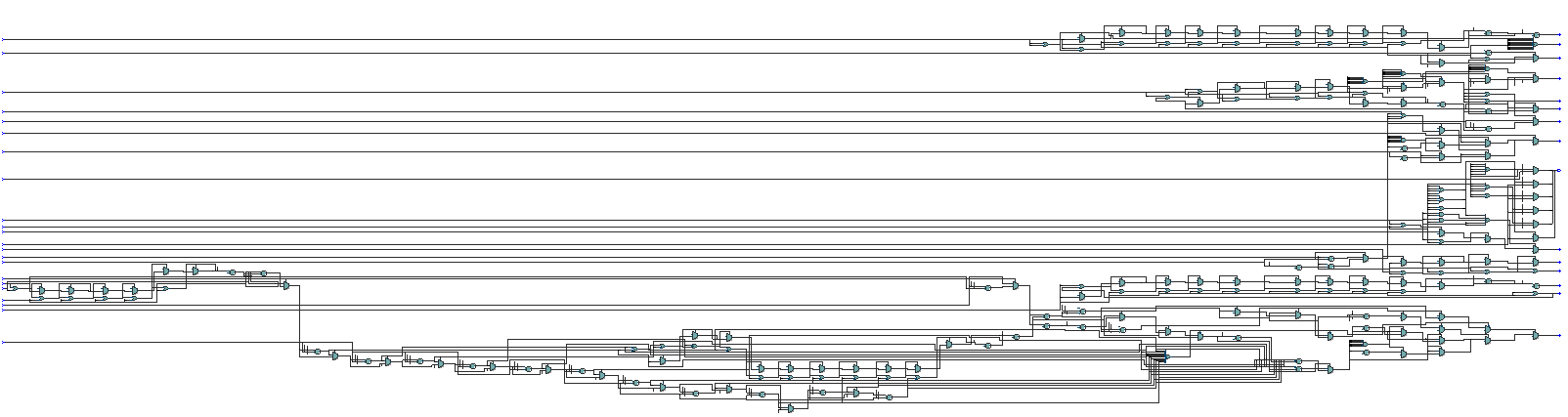




若我們把輸入加大成跟輸出一樣10bits，則圖形的密合度就會非常高，也可以看出每筆資料的輸入都有不同。

# RTL

此為Quartus II產生的RTL圖，因synplify加入float pak後編譯失敗尚未找出原因，所以附上為Quartus II的RTL。

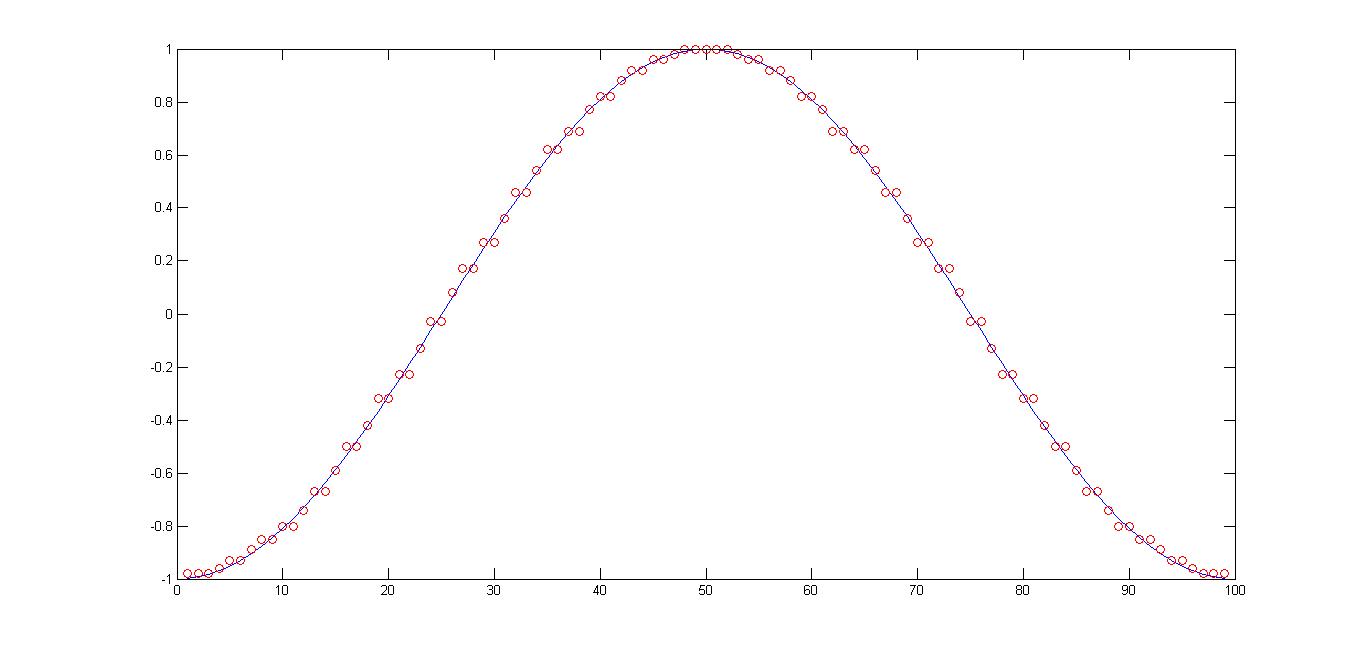


以上所有資料皆在專案資料夾底下的A1的資料夾裡面，Matlab讀檔程式則在專案資料夾底下的Matlab資料夾底下。

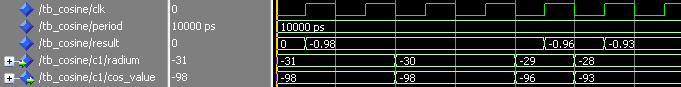
方法二 std\_logic\_vector

1. 實驗方法: 僅使用std\_logic\_vector型態，因為泰勒展開式的範圍限制，須在testbench將輸入的角度實數值，先減去使其數值範圍介於，將實數部分乘上一數值10，使其小數部分進位至整數-31~31，再以整數直接做泰勒展開式運算，運算後須保留小數部分，並在testbench當中轉換回實數，即可得到cos()。

# 二、ModelSim Simulation & Matlab

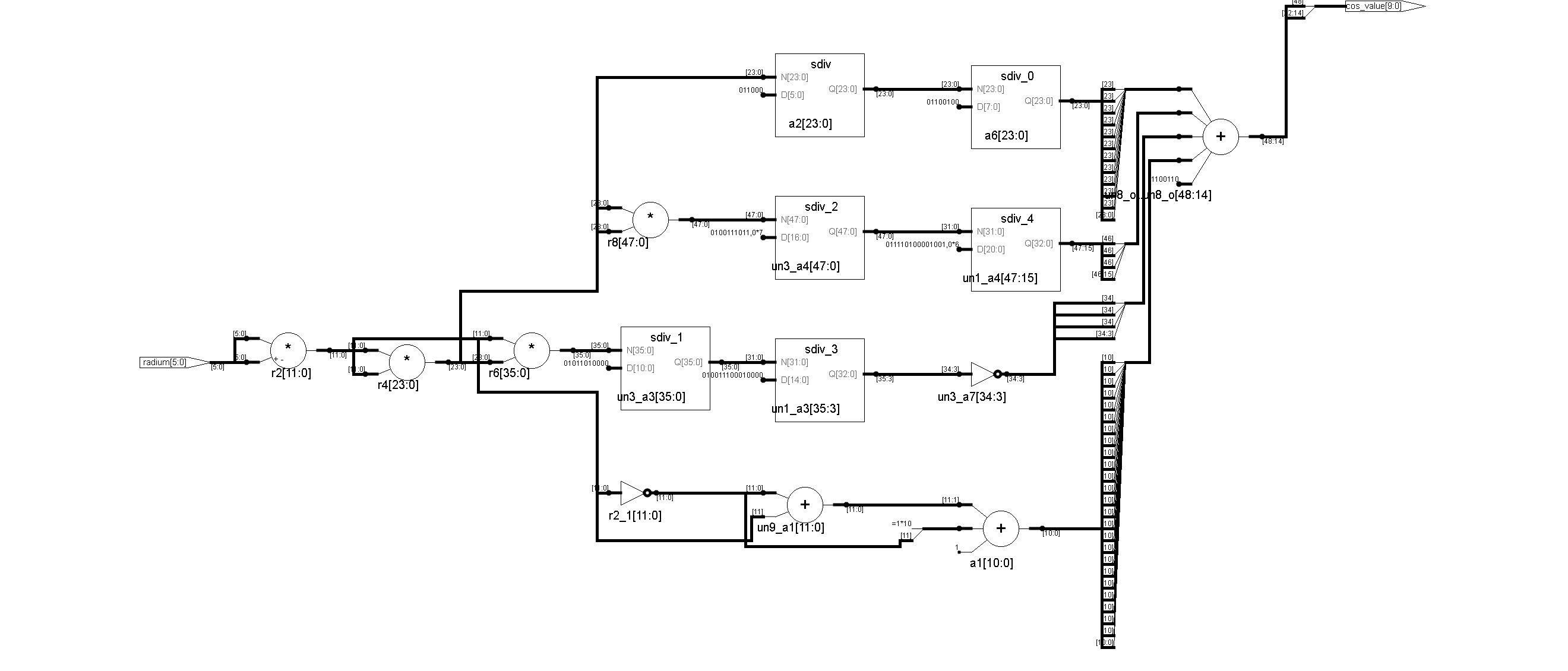


紅色點為系統輸出點，藍色線為Matlab cos圖形。相較於方法一而言，相鄰的點重複數值的個數較少，因為6 bits輸入的std\_logic\_vector，可以儲存-31~31，相當於小數第一位，如下圖所示，在bit數少的情況來說，std\_logic\_vector的效果是優於IEEE 754的小數處理方式。



# Synplify

此圖為Synplify產生之電路圖。



以上程式碼和Modelsim專案皆在A2的資料夾裡面，Matlab讀檔程式則在專案資料夾底下的Matlab資料夾底下，Synplify專案則在syn資料夾底下。