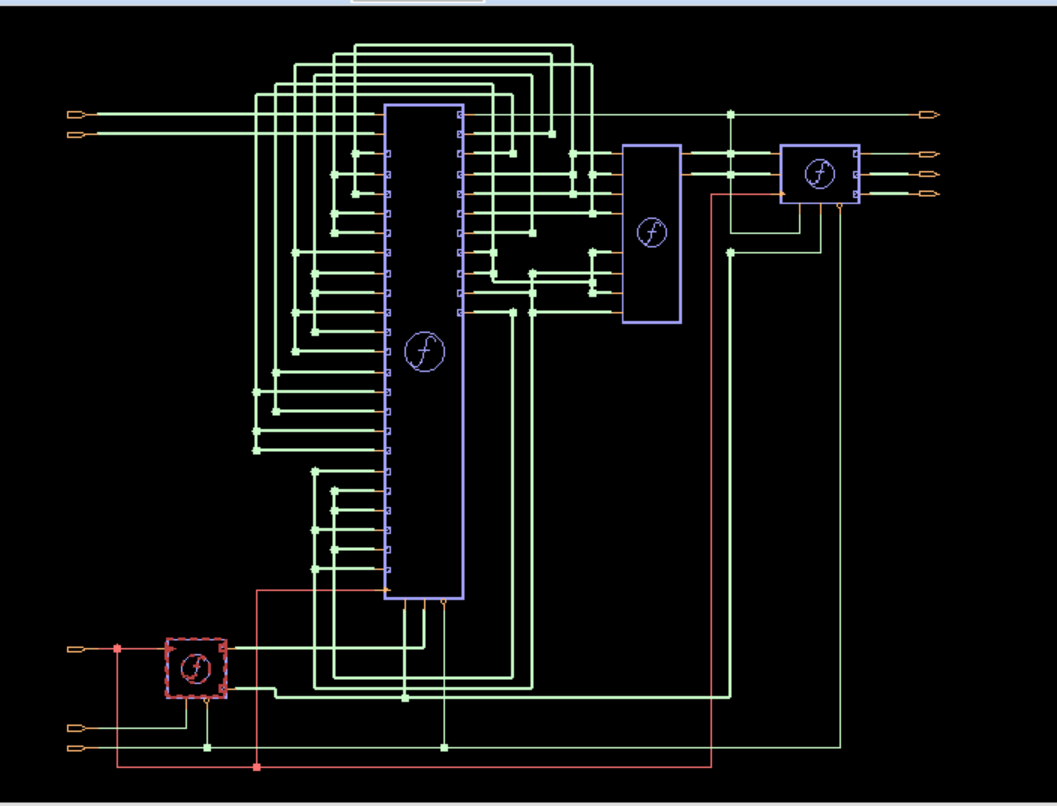
**Block Diagram**



output control

arithmetic operation

counter controlled opertations

shift hold

out\_image

out\_real

busy

outvalid

in\_image

in\_real

clk

invalid

rst\_n

**設計方法**

我用一個3bit的counter來控制當下要做的任務並用case去判斷，而其中要做的事就是讀input和吐output，因為我使用busy恆為0的方法去寫這次的作業所以outvalid會呈現一高一底的週期，而在低的時候就是我對我的matrix做shift (換column, row)，一邊計算一邊吐值；而讀值方面則是獨立掃到一暫存的array，避免新進來的input去干擾正在進行運算的數字，等到讀完完整的矩陣再傳到運算矩陣。

**心得**

這次我體會到其實面積也是會隨simulation time而變動的，所以不能只一味顧及面積的壓縮，而還要考慮到時間的調控，裡用兩者的乘積找到最佳解才是真正優化自己的電路!

**遇到困難**

這次主要遇到的困難為一開始第一筆資料還沒計算完out\_valid就開始拉起了，所以我就用一個8bit的reg對invalid做shift，確認第一筆資料已經充分算完才開始output，而後就依照前述的規律去做。

**A\*T優化過程**

我這次的A\*T優化過程大致如下:

area sim\_time (cycle time) A\*T

57160 5.0 ns 285800

57782 4.9 283131

58870 4.8 282576

60174 4.7 282817

61787 4.6 284220

可以看出4.8ns所產生的A\*T為最佳解，再大再小就會逐漸遞增

