積體電路設計 hw3　　姓名：鐘民憲

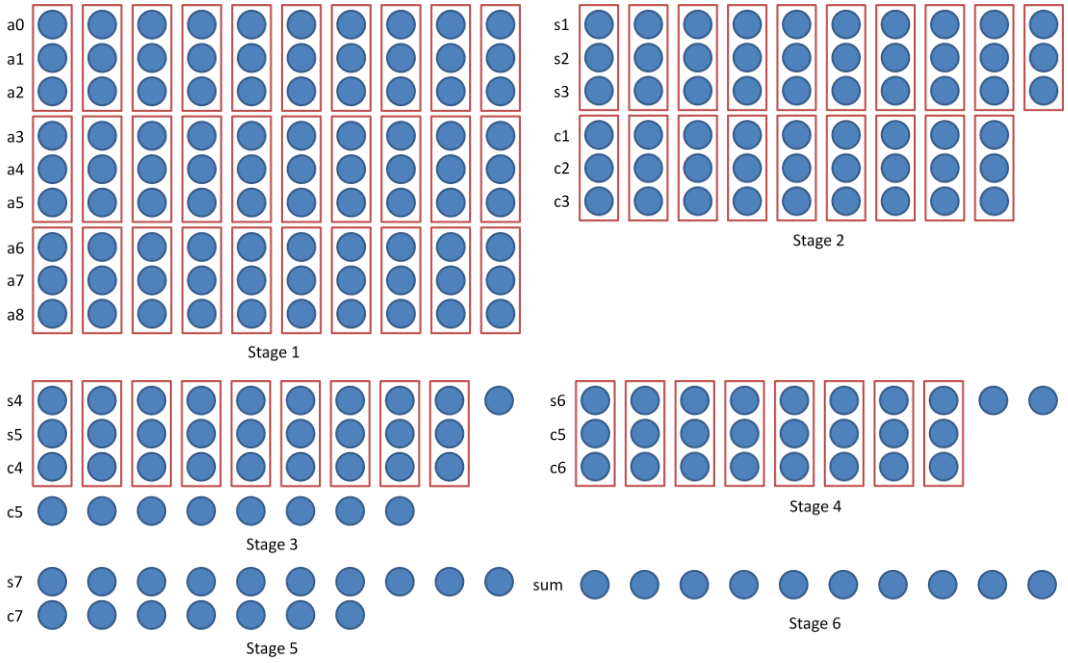
學號：b06901017

1. Adder part

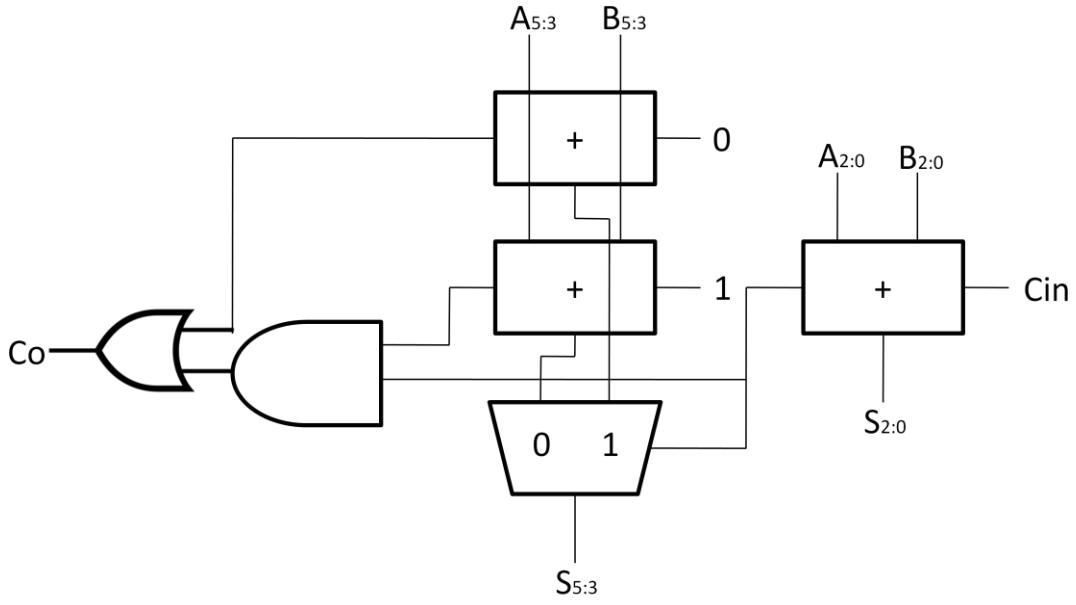
考慮九個 6bits 的 2's complement 數字相加後最高 overflow 到 10bits(31*9<2^9)，所以在做加法前先把每個數字都做 sign extension 到 10bits(i0->a0,i1->a1,…)

接下來我參考第八章上課投影片 p25 頁的方法使用 carry save adder 做加法：

(下圖中每個紅色方框代表一個 full adder)



做到 stage5 之後，接下來要處理兩個 7bits 數字的相加，由於第一個 bit 的 cin 為 0，所以第一個 bit 用 half adder 來做，剩餘的 6 個 bits 就用如下圖的 carry select adder 來相加(圖中+的部分為 carry ripple adder)，由於在前面已經處理好 overflow 的部分，所以 Co 可直接省略。



2. Divider part

比較棘手的反而是做除法，因為 1/9=1/16+1/32+1/64+1/1024+1/2048+1/4096…，完全不知道要到多少精確度才夠，而且由於做越多個數字或越多 bits 的加法，運算速度就越慢，所以只能一個一個慢慢嘗試，儘可能在加最少數字及 bits 的情況下達成 100%正確率。

我一開始只有加前三個數(共 12 個 bits)，發現當 sum 很小或很大的時候，完全沒辦法藉由小數點以下的位數去區分應不應該進位，精確度太低。

接著嘗試只加前四個數(共 16 個 bits)，然後用小數點下六位數的大小去判斷四捨五入，若 sum 為正，只要小數點後的數字>=0.011101 就進位，反之則否；若 sum 為負，只要小數點後的數字>=0.100010 就進位，反之則否。雖然跑出來的 latency<6ns，可是 10000 個測資中有 23 個錯誤，真是可惜。

最後加到第五個數 (共 17 個 bits)，發現只要看小數點後一位是不是 1 來決定進位就行了，不過由於要加的數字跟 bits 數都多了一位，latency 變成只能<7ns，稍微慢一點，不過正確率就能達到 100%了。

Divider part 做加法的方式跟 Adder part 相同，都是先用 carry save adder 處理到只剩兩個數字後，再用 carry select adder 相加，最後得到的 17bits 數字中，前 6 個 bits 為平均，第 7 個 bits 決定是否進位，進位的部分由於只需要加 1，所以使用 carry ripple half adder 來做即可。

3. Discussion

(1)在我的 code 中，處理兩個數字相加的 full adders 分別有 3bits, 4bits, 6bits, 14bits，對這些 full adders 進行測試後發現 3bits 跟 4bits 使用 carry ripple adder 的速度較快，而 6bits 跟 14bits 則是使用 carry select adder 比較快。

(2)使用單純 carry ripple adder 的速度比使用 carry-ripple pg logic 的速度快。

(3)在詢問其他同學後發現全部都用 carry ripple adder 做加法的速度跟用 carry save adder 加 carry select adder 的速度差不多。可是根據老師上課的講法，carry ripple adder 應該要是最慢才對，不然也不需要去設計其他 adder 出來，因此覺得模擬出來 latency 幾乎沒有差別是件相當奇怪的事情，猜測是在 lib.v 中 assign 每個 gate 的 time latency 不夠正確的關係。

(4)Half adder 的速度比 Full adder 快上不少，所以能用 half adder 的地方就盡可能用 half adder，就算是只多 1 個 bit 用 half adder 也能看到速度加快。

(5)大致上覺得根據自己設計的架構，小於 7ns 就是極限了，除非採用其他的 adder，不然應該很難突破 6ns。