

(1) 我是直接寫 32bit，當加法時，若 $a[31], b[31]$ 都等於 1 (即 sign bit 都為正)，但相加後結果 $result[31]=0$ ，即有 overflow 產生，因為正的與正的相加不可能為負，同理若 $a[31], b[31]$ 都等於 0 (即 sign bit 都為負)，但相加後結果 $result[31]=1$ 也視為有 overflow 產生；當減法時，若 $a[31]=1, b[31]=0$ ，因 $a-b=a+(\sim b+1)$ ，此時 $a-b$ 可視為負數 a 加負數 b ，若 $result[31]=0$ ，同加法為正的與正的相加變負

的，即有 overflow 產生，同理 $a[31]=0, b[31]=1, result[31]=1$ 時，也視為有 overflow 產生。

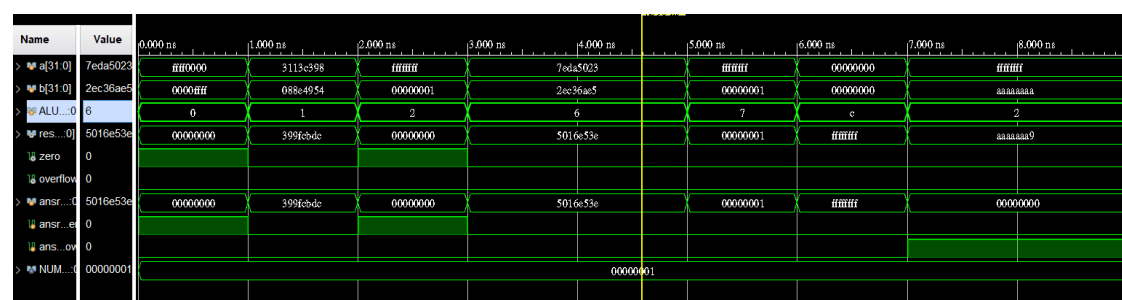
(2) 由 1bit ALU 來看，ALU control signal 由(a_invert,b_invert,operation)組成，從 sub=0110 來看，a 不需要 invert 所以第一位為 0，b 需 invert 第二位為 1，而 operation 的部分，00 為 and，01 為 or，10 為 add，而 sub 為 add 第三第四位為 10，組合成 0110；nor=1100 即 a,b 都需 invert，operation 為 and，組合成 1100。

(3) 若 assign 不同的 value，需要去修改 1bit ALU 使其符合自己的 assign，否則會導致得到的數值錯誤，例如將 sub=1010，即可將 a_invert 與 b_invert 的位置調換。

(4) false，雖然 register file is both read and written on the same clock cycle,他有一些方法可以管控，讓他不需更多的 register file 來管控，例如: Edge-Triggered Writes, register file 更新當 rising or falling edge of the clock signal，確保此部會互相干擾；Control Signals and Timing, 利用 enable/disable read and write operations 去控管什麼時候要 read 什麼時候要 write.

3.

32bit ALU



我加入一組 $a=0xffffffff, b=0xaaaaaaaa$ 運算為相加，用來測試當產生 overflow 時的結果。

4,5.

寫的過程中遇到的問題比較少，大部分都是因為語法不熟悉，其他經過查資料後都大致能想到怎麼做。謝謝助教的詳細說明與分成多個 step 帶我們慢慢拼湊程式。