**積體電路電腦輔助設計概論**

**CAD LAB2**

**Scheduling**

**實驗日期： 2019/04/24**

資工110

B063040061

陳少洋

**(一) 實驗內容說明**

在課堂中，我們有學到各種不同的scheduling的方法。其中，應用List Scheduling為這次實驗的主要內容。

以List Scheduling的方式來做到resource constrained的目的，做法為先算出整個運算的longest path，並將input已經ready的operation分別放入ready list ( mul, add )，再以longest path為priority function去選擇path最長的優先運算，成為critical-path list scheduling。

**(二) 實驗過程說明**

**主要過程：**

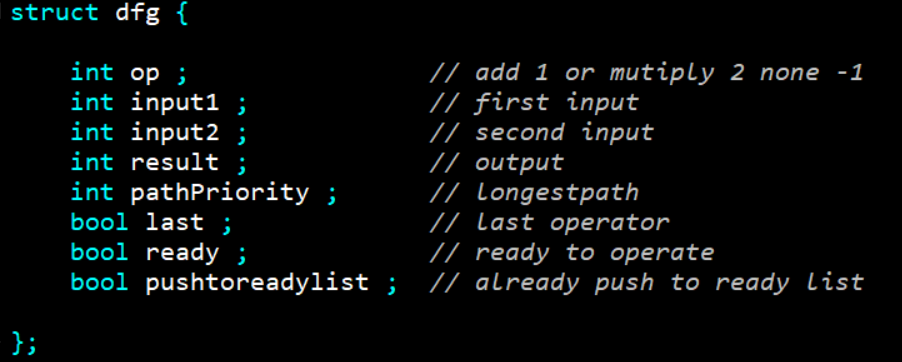
一開始想不太到要用什麼資料結構來完成這次的實驗，最初選用vector一筆一筆存資料再去作分析，但到了要算Longest Path的時候就出了很大的問題，後來改用SIZE固定200大小的陣列，來每個operation number來當陣列index，才解決看不到inpu1、input2是否ready的情況。

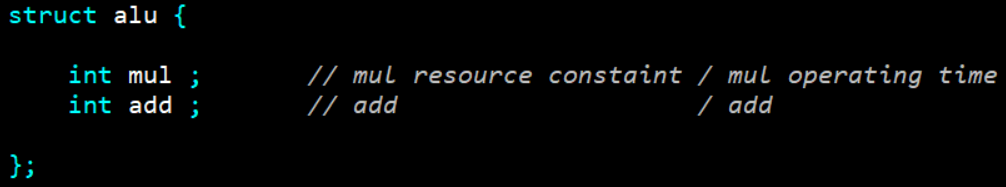
第二部分就是做LongestPath的運算，我是從整個運算的最後一個result( 不會有其他運算拿來當input )，往上做遞迴，每次都挑需要時間比較長的部分一個一個往上增加path長度，作為priority的依據。

最後一部分就是實作List Scheduling，主要是以for迴圈計算step count，直到所有的operation全部做完，而for迴圈中最主要的工作就是，將ADD/MUL的input1/2皆ready可用的operation分別推入ADD/MUL 的ready list( vector )，經由resource constrained的個數，放入閒置中的operator，最後再做運算倒數，如果倒數至0就把這個result設為ready可提供別人使用( ADD/MUL皆分開做 )。For迴圈結束後則return stepCount，就是最後所需要花費的時間。

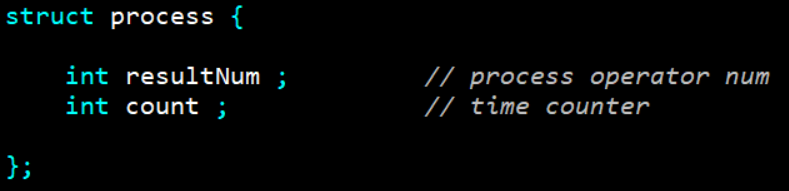
**程式碼簡要說明：**

**主要使用3個struct：**

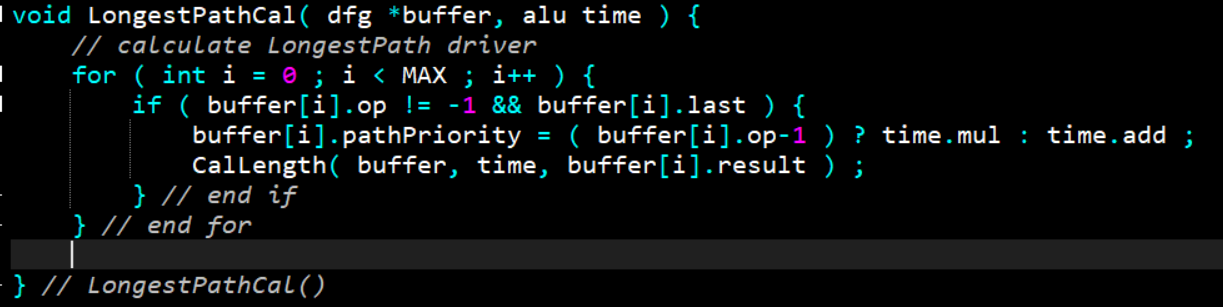
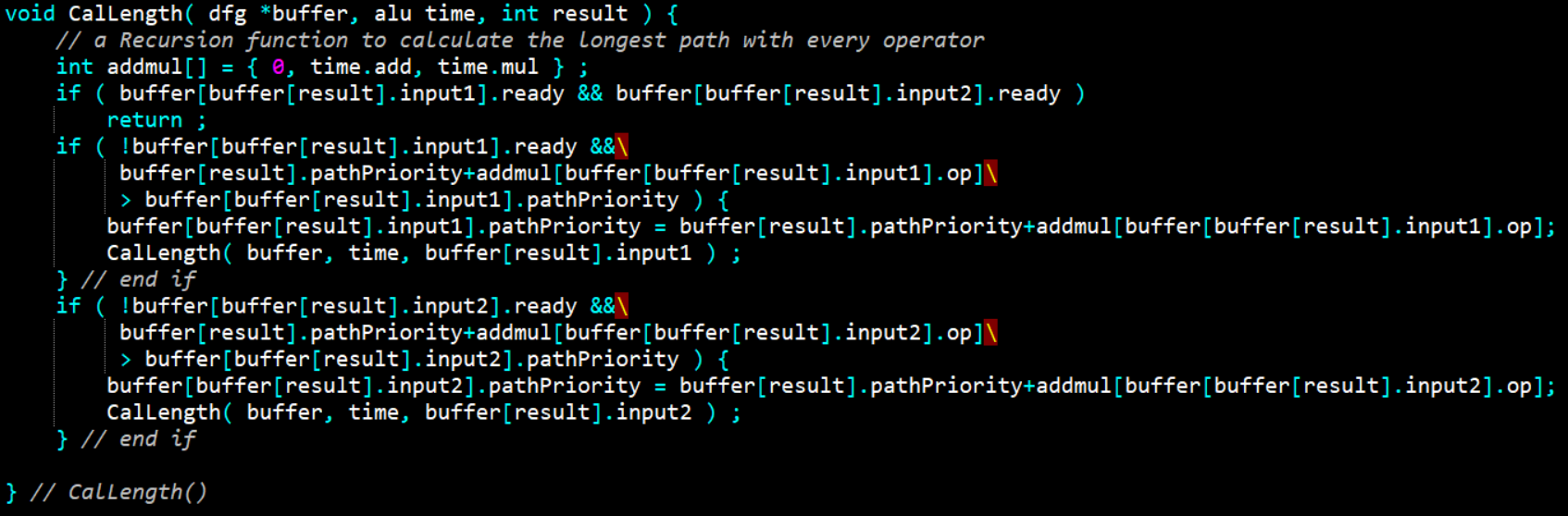
****

****

* dfg最主要的架構(記錄所有內容)
* Alu紀錄有幾個operator / 紀錄要用的時間(兩種用途)

****

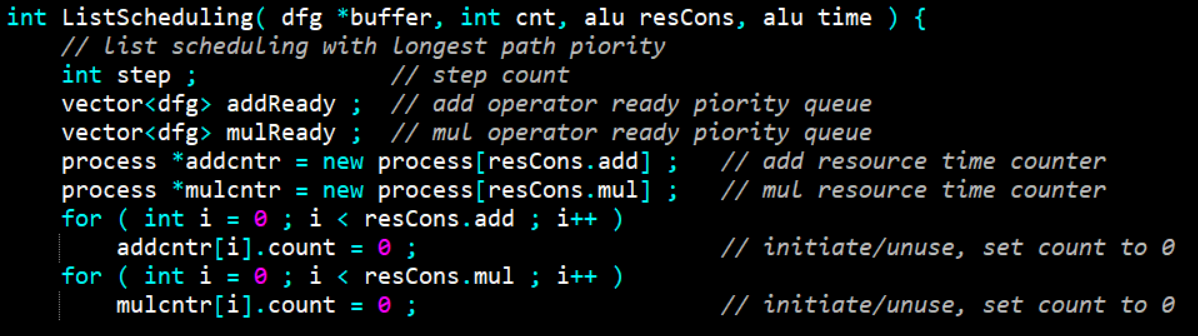
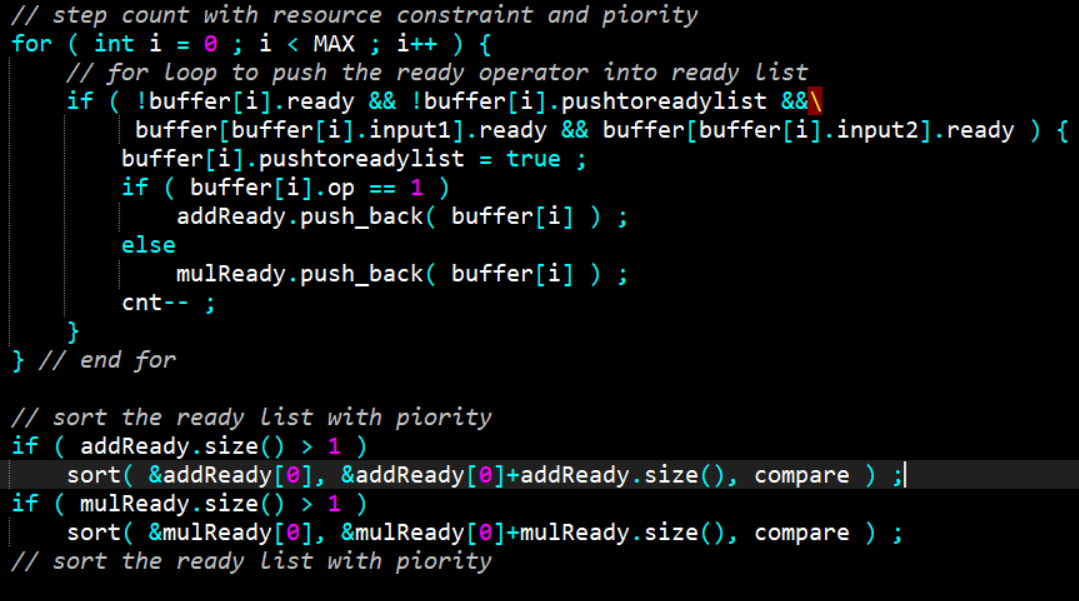
* Process紀錄目前運算中的operation編號、及其還需要多少時間能完成這個運算

** 計算LongestPath Priority：**

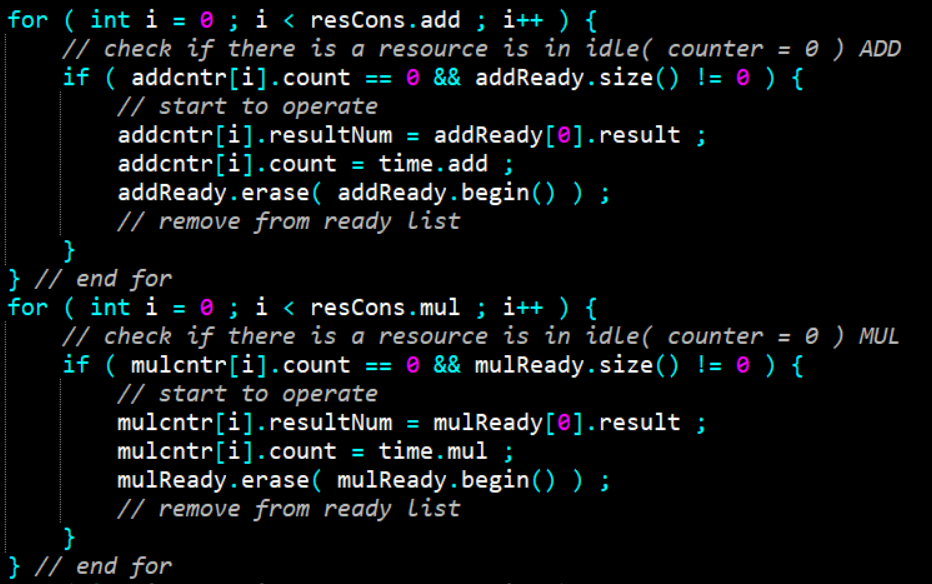
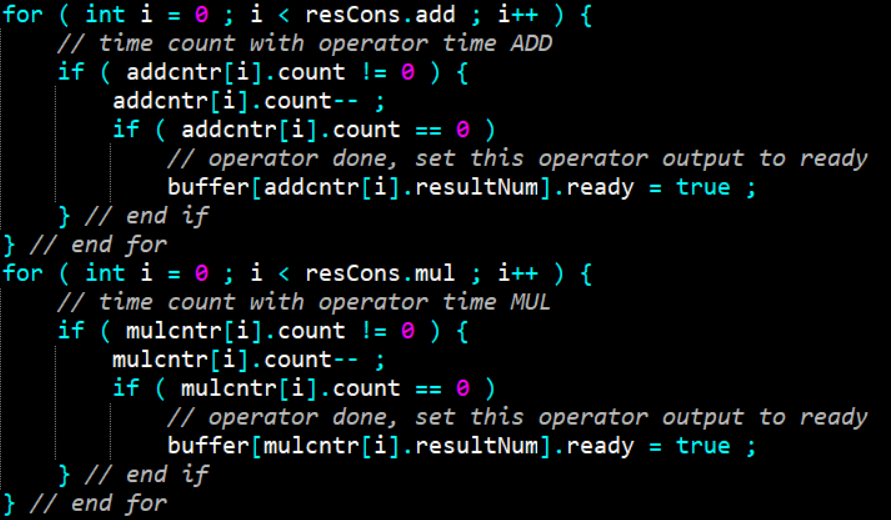
其中每次都去看input1跟input2兩個都ready的話，代表這是最一開始的operation，也就是base case。但只要其中一個不是最上層的operation就要繼續做recursion。接下來，每次都去檢查目前的pathPriority加上input1(or2)所花費的運算時間，是否大於input1(or2)目前所計算到的pathPriority，藉由這個檢查，就可以選擇最長的那個運算時間，得到最後的longest path。

* 上圖為CalculateLongestPath的driver / 下圖為recursion計算長度

**List Scheduling：**

****

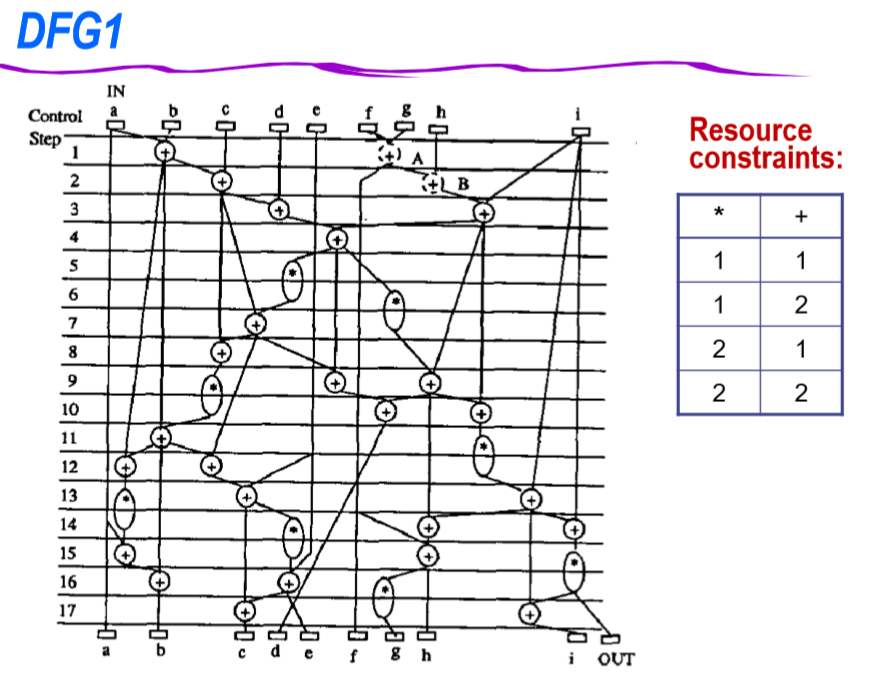
* List Scheduling // Step counting，推入ready list，並以pathPriority排序。
* List Scheduling 事前宣告

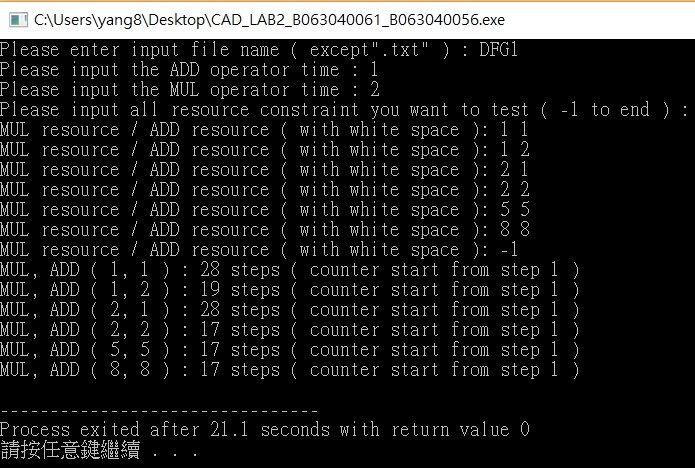
****

* List Scheduling // for迴圈結束則return step count，最終結果
* List Scheduling // 上為ADD下為MUL，把設置的time-1，若time-1為0，表示完成operate
* List Scheduling // 上為ADD下為MUL，檢查有沒有閒置的operator，有就丟進去並設置time

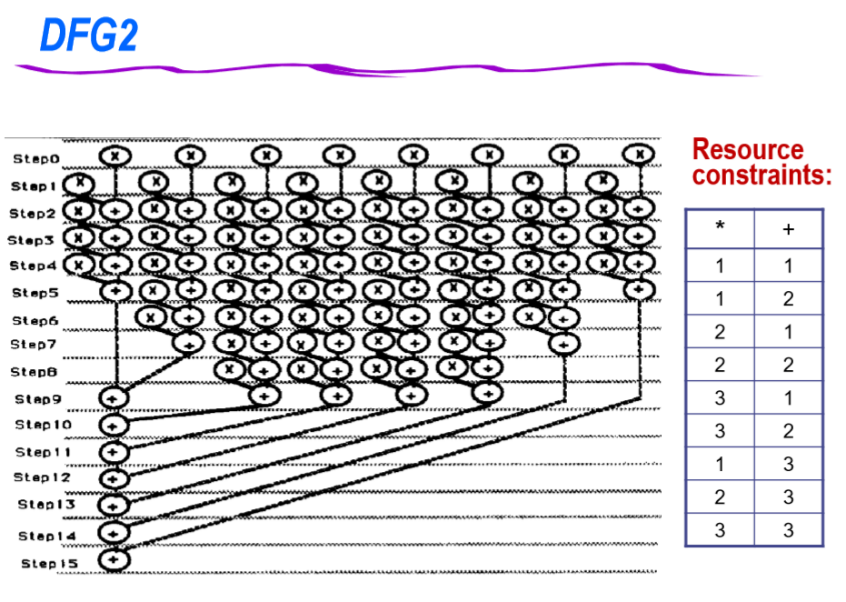
**(三) 實驗結果分析說明**

**測資一：**

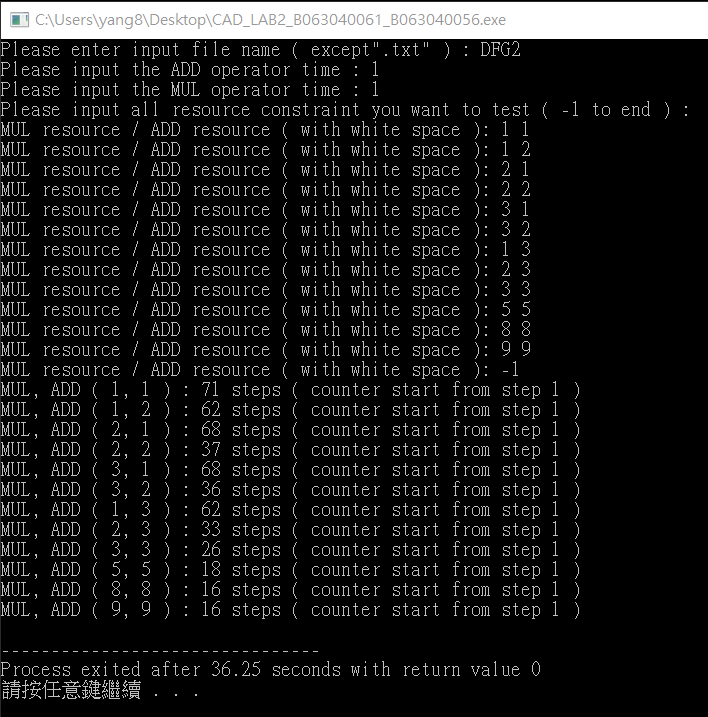
****

**** 總共有4個資源限制，我多測了兩個，來證明critical path是17 steps，而老師PPT的圖為resource constraints( mul = 2, add = 2 )為17 steps。

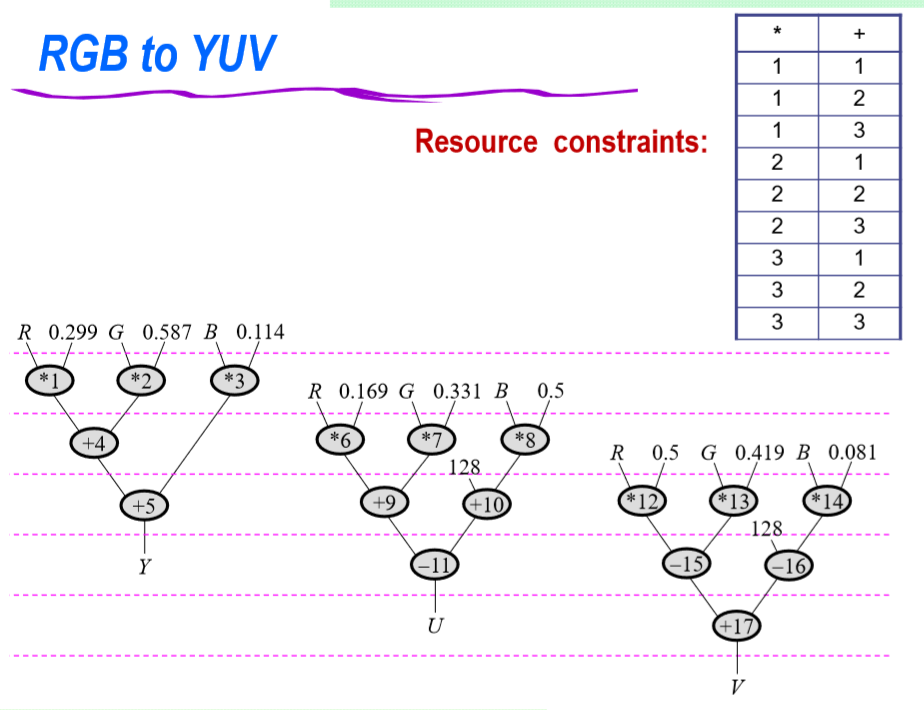
**測資二：**

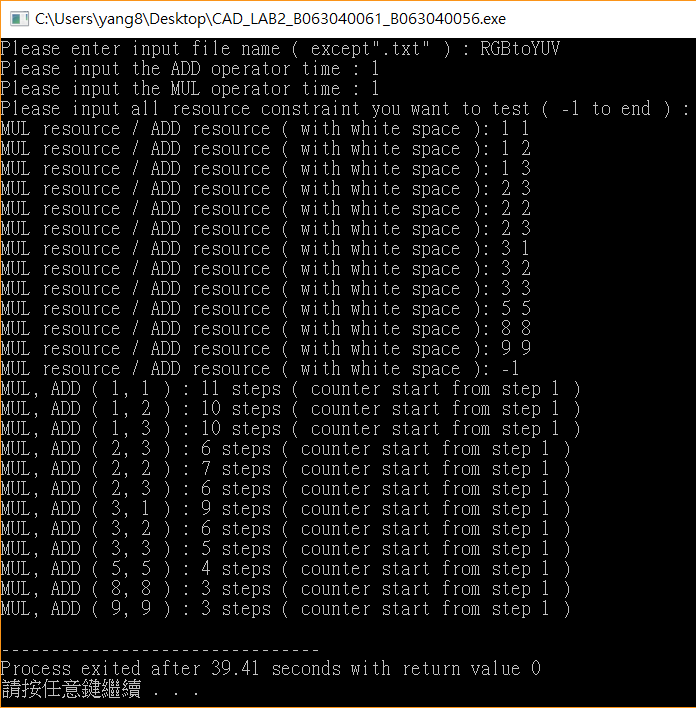
**** 我一樣多試了兩個resource constarints，得到critical path為16 steps，而左圖為resource constraints( mul = 8, add = 8 )為16 steps。

( 0 ~ 15 = 16 )

****

**測資三 ：**

一樣多試了兩個resource constarints，得到critical path為3 steps，而左圖為resource constraints( mul = 3, add = 3 )為5 steps。****

****

**(四) 實驗心得**

第二次的積體電路電腦輔助設計實驗，在一開始聽老師講解scheduling的時候，概念大致上都了解也聽得懂，但真正要實作時，卻又不知道該如何下手。但也因為有這次實驗，我又重新再了解一次List scheduling的精隨，最後也創造出自己想法中的資料結構，雖然過程中並沒有太容易，但大致上都能一一克服，也讓我更加對這些scheduling的方法有更進一步的認識。

事實上，理解一個完整的概念與實作出一個完整的架構還是有一些落差，時常與我所想像的做法不同，但慢慢增加東西而完成後，就覺得還蠻有趣的，也有很多不同的做法能達成一樣的目的，希望未來需要我做其他更進一步的軟體輔助硬體設計下，這堂課所學的東西能夠派上用場。