**積體電路電腦輔助設計概論**

**CAD LAB3**

**Allocation**

**實驗日期： 2019/05/15**

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allocation的方法，例如Greddy Constructive Approach、Clique Partioning、Iterative Refinement Approach等等。其中，我們應用Left-Edge Algorithm為這次實驗的主要內容方式。

以Left-Edge Algorithm的方式來做到Allocation的目的，來確保我們可以運用最少的register file個數。這個演算法的最大的限制就是，在同一個register file內，不能有重複life time的運算資料，也就是說我們只要記錄運算data的life time起始與結束就能在眾多data抓出可以共用register file的data life time。

**(二) 實驗過程說明**

**主要過程：**

**Step1** 讀入所有運算的名字、起始時間與結束時間

**Step2** 以起始使間去做運算的分類並給予排序

**Step3** 從第1個Register開始抓不衝突的資料進入Register vector

**Step4** 如果起始時間比暫存器目前使用時間大就可以共用此Register

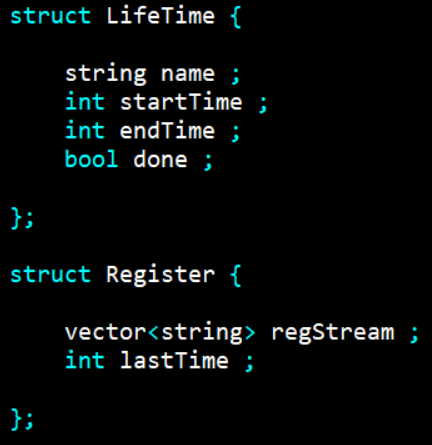
**Step5** Register last time設為此運算結束時間、並記錄其名字

**Step6** 換下一個Register抓data進來共用

**Step7** loop step4~6直到所有data都有分配到Register

**程式碼簡要說明：**

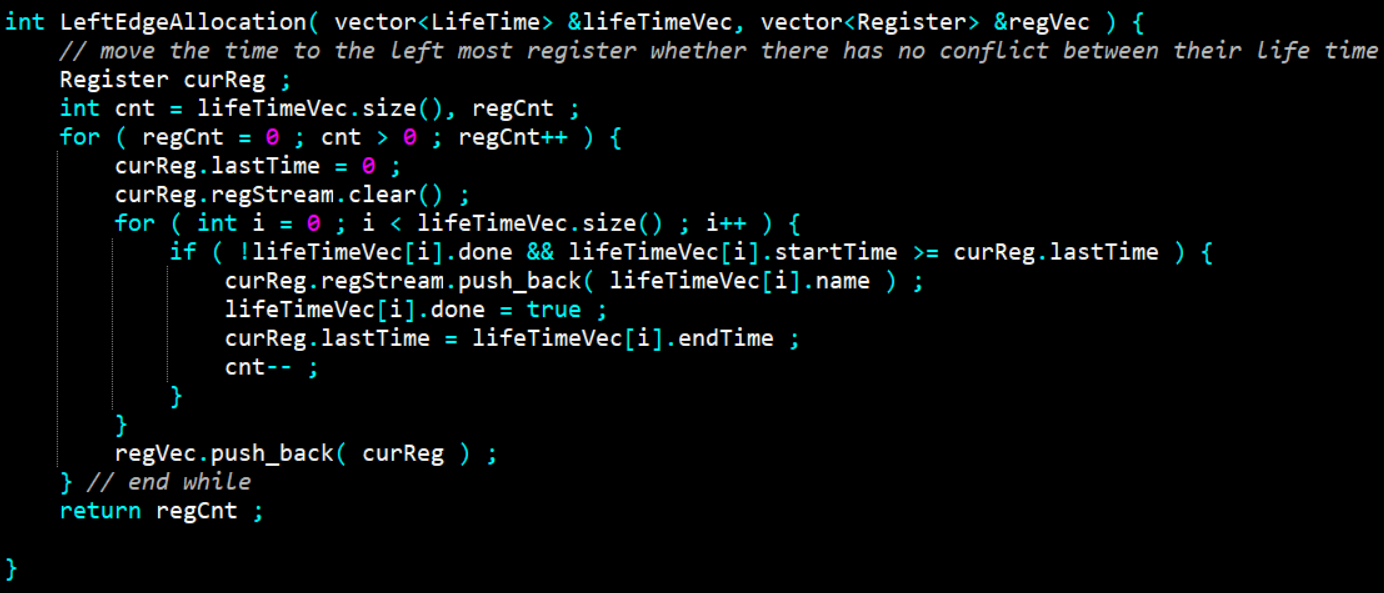
主要使用2個struct：



◤Life Time代表此運算的生成時間直到其被用掉的時間

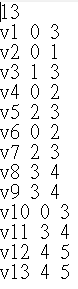
◤Register代表目前暫存器所存內容

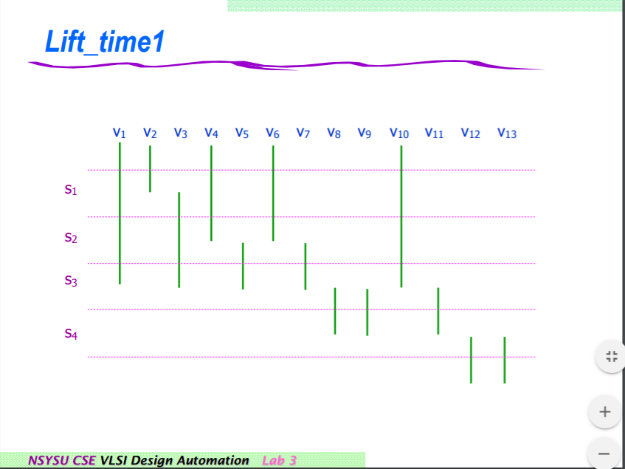
Left-Edge Allocation Algorithm：

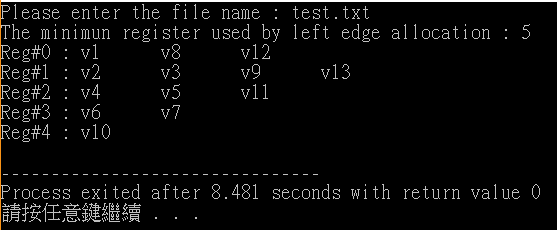


* LeftEdgeAllocation 一個for迴圈就代表需要新增一個新的Register來使用。將可以共用的資料存入當前Register中，計算出最少需要用到的Register以及共用資訊。

**(三) 實驗結果分析說明**

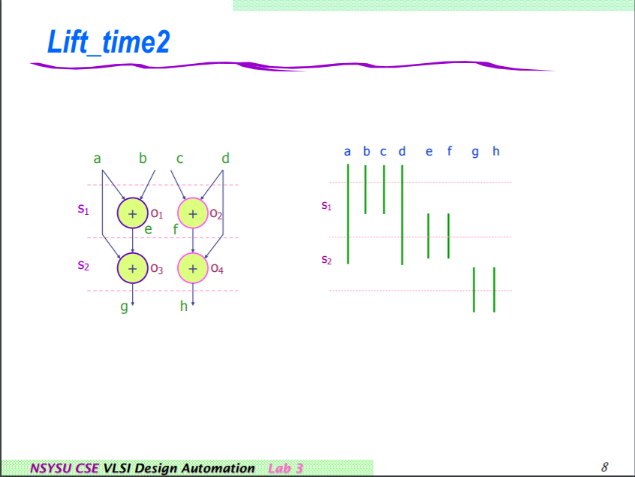
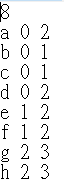
** 測資一：**

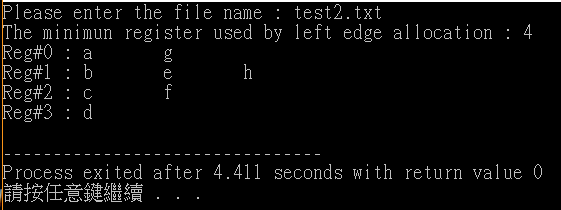
****

****

我們可以由上圖得知各個Register所需要存取的運算，並將其往左移動歸類在同一個Registerd可以發現他們的life time並不衝突。

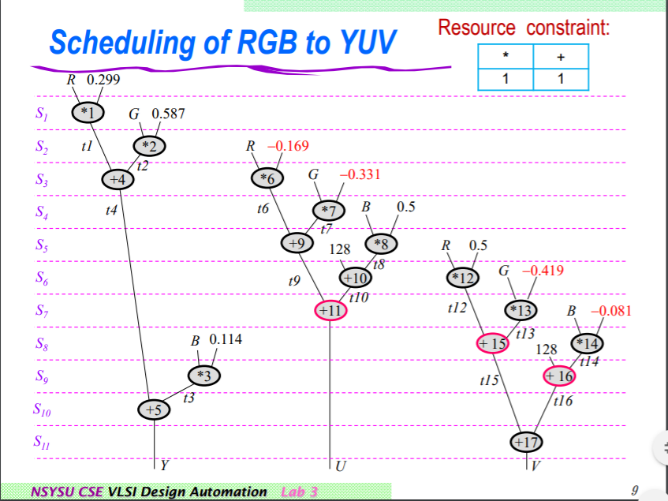
**測資二：**

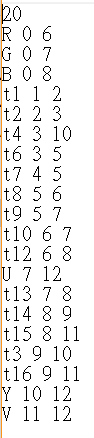
****

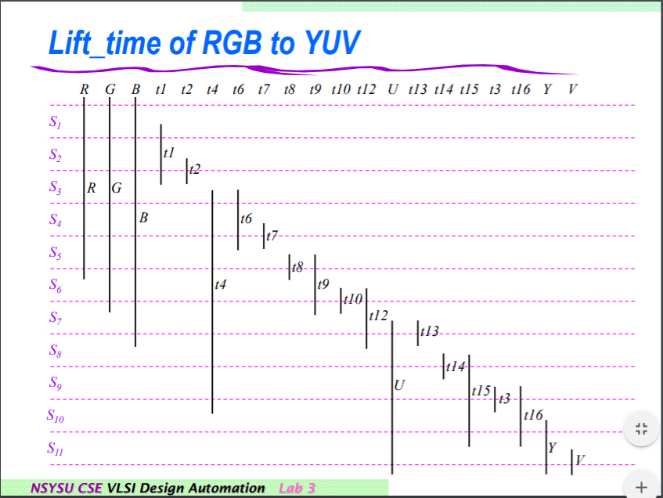
****

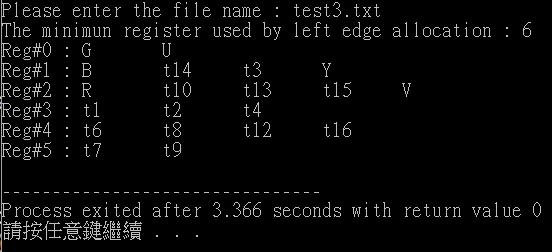
我們可以由上圖得知各個Register所需要存取的運算，並將其往左移動歸類在同一個Registerd可以發現他們的life time並不衝突。

**測資三：**

****

****

****

****

**Reg#0：G( 0~6 ) -> U( 7~12 )**

**Reg#1：B( 0~8 ) -> t14( 8~9 ) -> t3( 9~10 ) -> Y( 10~12 )**

**Reg#2：R(0~6)->t10(6~7)->t13(7~8)->t15(8~11)->V(11~12)**

**Reg#3：t1( 1~2 ) -> t2( 2~3 ) -> t4( 3~10 )**

**Reg#4：t6( 3~5 ) -> t8( 5~6 ) - >t12( 6~8 ) -> t16( 9~11 )**

**Reg#5：t7( 4~5 ) -> t9( 5~7 )**

**(四) 實驗心得**

這次的實驗比較容易理解，Left-Edge Algorithm算是一個很好理解的Allocation的方法，也因此這次能比較快速的寫出這個演算法所要求的條件，只差在需要自己想好想要使用的架構，所以我分了兩種架構來區別不同的運作。

雖然說這次實驗不難，但這只是整個模擬的一個小部分，且這只是其中一種實作方法，老師上課也講了許多不同的Allocation方法，我認為自己也都有吸收理解，雖然理解不一定能轉換成程式碼，但這次透過實驗內容，讓我又對Allocation的方式有更深入的認識，希望未來有機會在實作這部分的話，能更快的對整個完整的架構上手。