# **實驗二 ARM Assembly II**

**0610032 林仲偉**

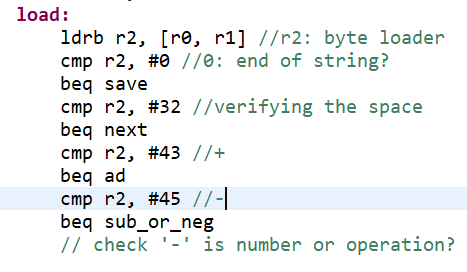
1. **What** 實驗要做什麼

熟悉基本ARMv7組合語言語法使用。

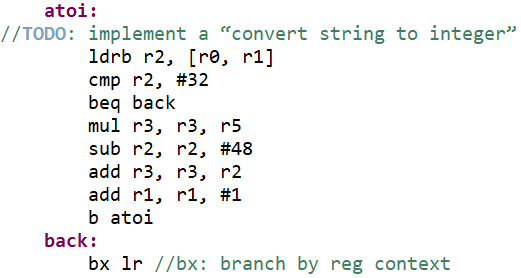
1. **How** 實驗要怎麼做
2. **Postfix arithmetic (50%)**

**操作 stack 來完成 postfix 的加減法運算**

每次都讀入一個字元(R2, in asci value)，依照該字元決定操作。



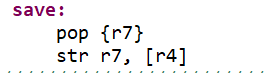
* **R2不觸發上述任何branch: 默認為數字，把ascii轉int後再push**

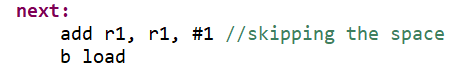


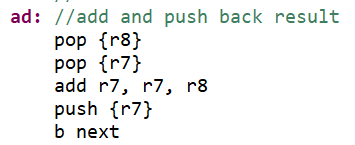
讀到空白字元時branch back，否則就一直讀新值&舊值<=舊值\*10+新值

記得ascii要 -48 就能轉成單一字元表示的數字( 0:9 = ascii(48:57) )。

* **R2=0: 讀到end of string，就把stack內結果(應該只會剩一個)pop並存入result。**

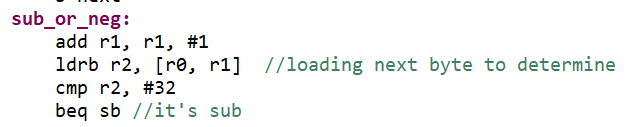


* **R2=32: 讀到space，直接繼續讀下一個。**
* **R2=43: operation +: pop兩個數字做運算再push**

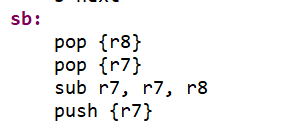


* **R2=45: 需要先看是operation還是sign:**

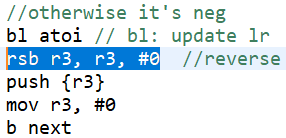
再load下一個字元，如果是空格就是operation; 否則是sign



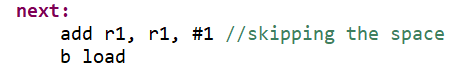
Operation:和+一樣，pop兩個數字做運算再push

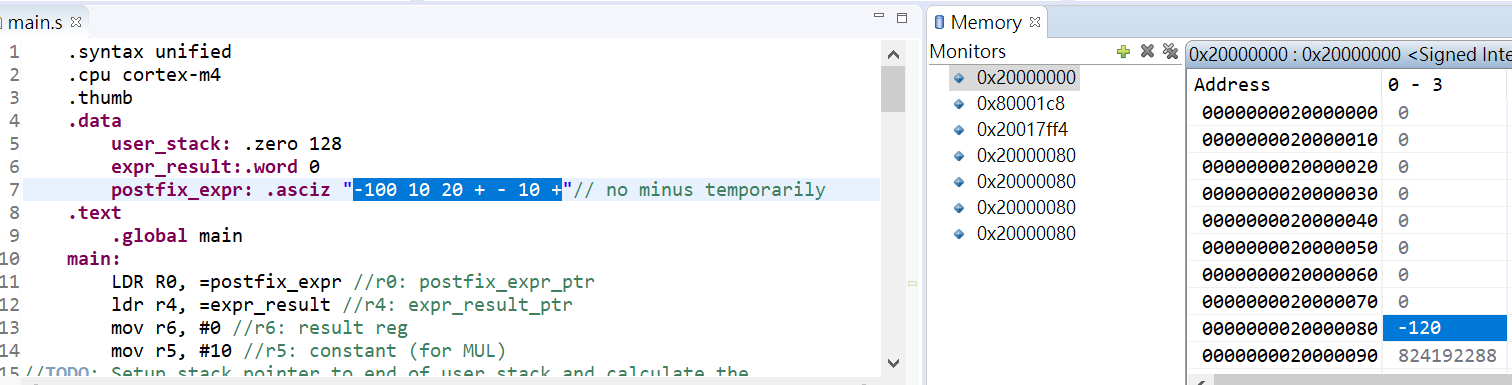


Sign: 和把ascii轉int後再push一樣，只是多了一個RSB來反轉正負



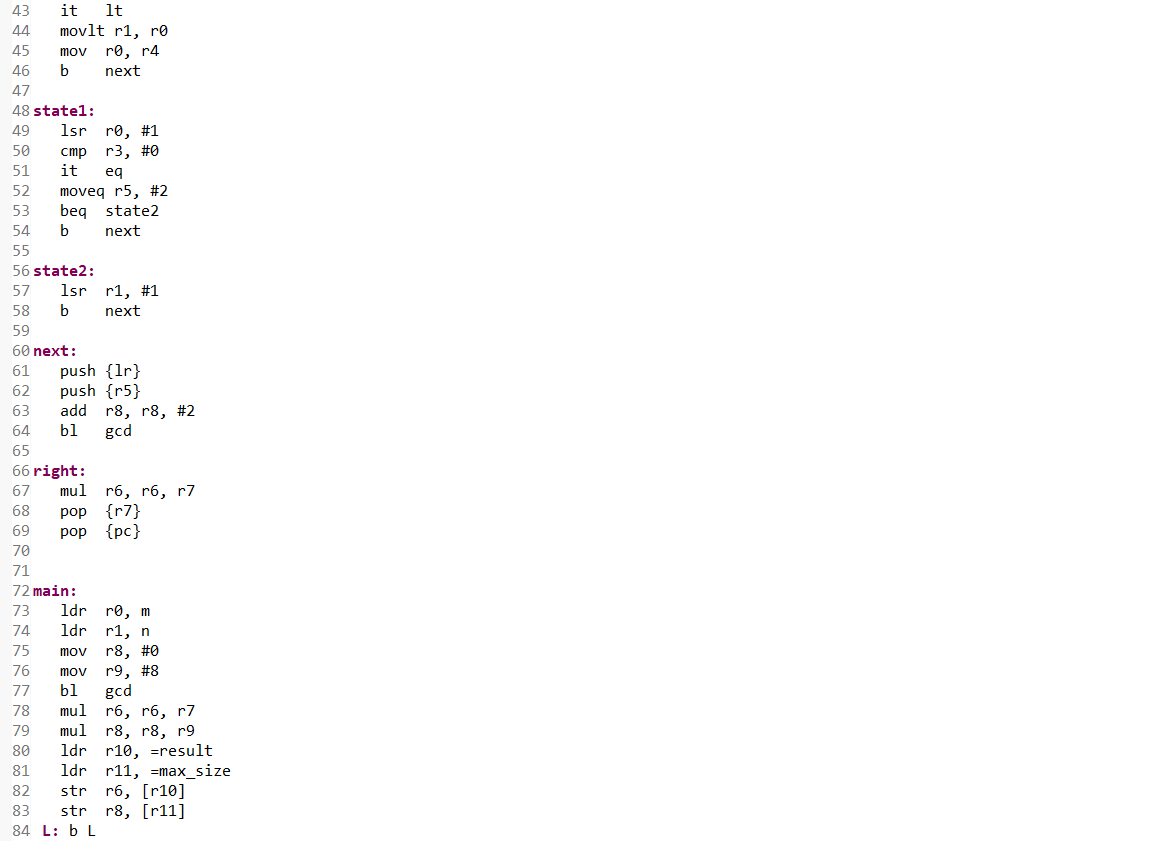
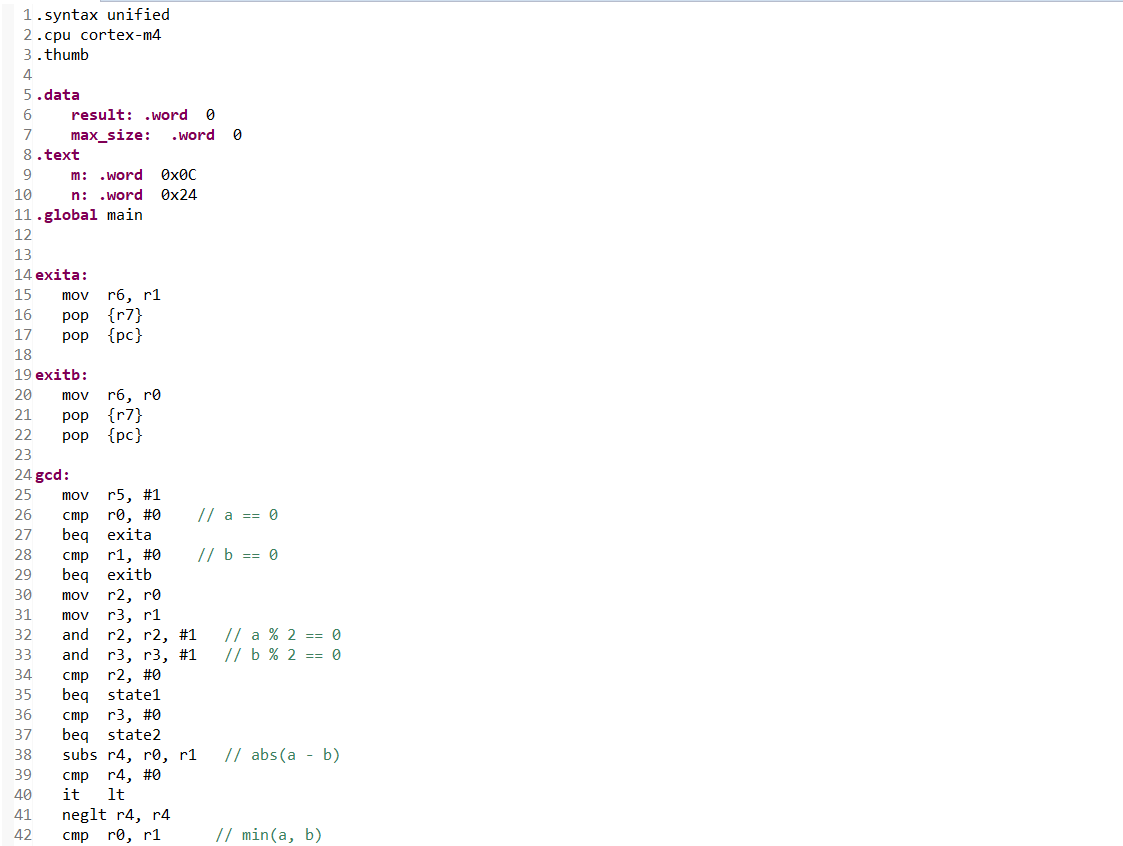
* **每個區塊最後都要跳下一個字元: branch to “next”**



 最後結果:

1. **求最大公因數並計算​最多​用了多少 stack size (50%)**

* **Code:**



* **主程式部分**

1.本程式使用了r0 ~ r11的register，功能如下。

|  |  |
| --- | --- |
| r0 | 儲存並運算m的值 |
| r1 | 儲存並運算n的值 |
| r2 | m % 2 的結果 |
| r3 | n % 2 的結果 |
| r4 | abs(m, n) |
| r5 | 要儲存的乘數 (1或2) |
| r6 | gcd的結果 |
| r7 | 被取出的乘數 |
| r8 | stack size |
| r9 | 儲存數字8 |
| r10 | 儲存result的位置 |
| r11 | 儲存max\_size的位置 |

2.將m和n值讀入r0和r1，跳至gcd進行運算。

3.Stein’s algorithm可被判斷式分為6種情況，其中4種須再次呼叫gcd，在此介紹根據判斷式所跳至的區塊及其需進行的工作。

**若r0 == 0**，跳至exita: 將r1的值放入象徵result的r6，將之前存的

位置由stack pop出至pc。

**若r1 == 0**，跳至exitb: 將r0的值放入象徵result的r6，將之前存的

位置由stack pop出至pc。。

**若r0 % 2 == 0**，跳至state1:將r0除以2，判斷r1是不是同時也是偶數

，若是，將乘數r5設定為2，跳至state2，若非，則跳至next。

**若r1 % 2 == 0**，跳至state2:將r1除以2，然後跳至next。

若以上情況皆不符合，則將r0和r1設為abs(r1-r0)和min(r1,r0)

abs(r1-r0)的作法:將兩者相減的結果與0比較，若小於0則使之neg。

min(r1,r0)的作法:將兩者相比，若r0較小則移入r1，反之則不處理。

**next**:進入此區塊表示需進行下次gcd。將lr及乘數r5 push至stack，並將紀錄stack size的r8加2，利用bl重新跳至gcd。

**right**:由於right位於next下方，因此當exita或exitb pop出位置至pc後，便會跳至此區塊。將r6與乘數r7相乘，再pop出下次需要的位置和乘數。

4.由於最初紀錄的乘數與返回main的位置同時被取出，未被計算到，因此返

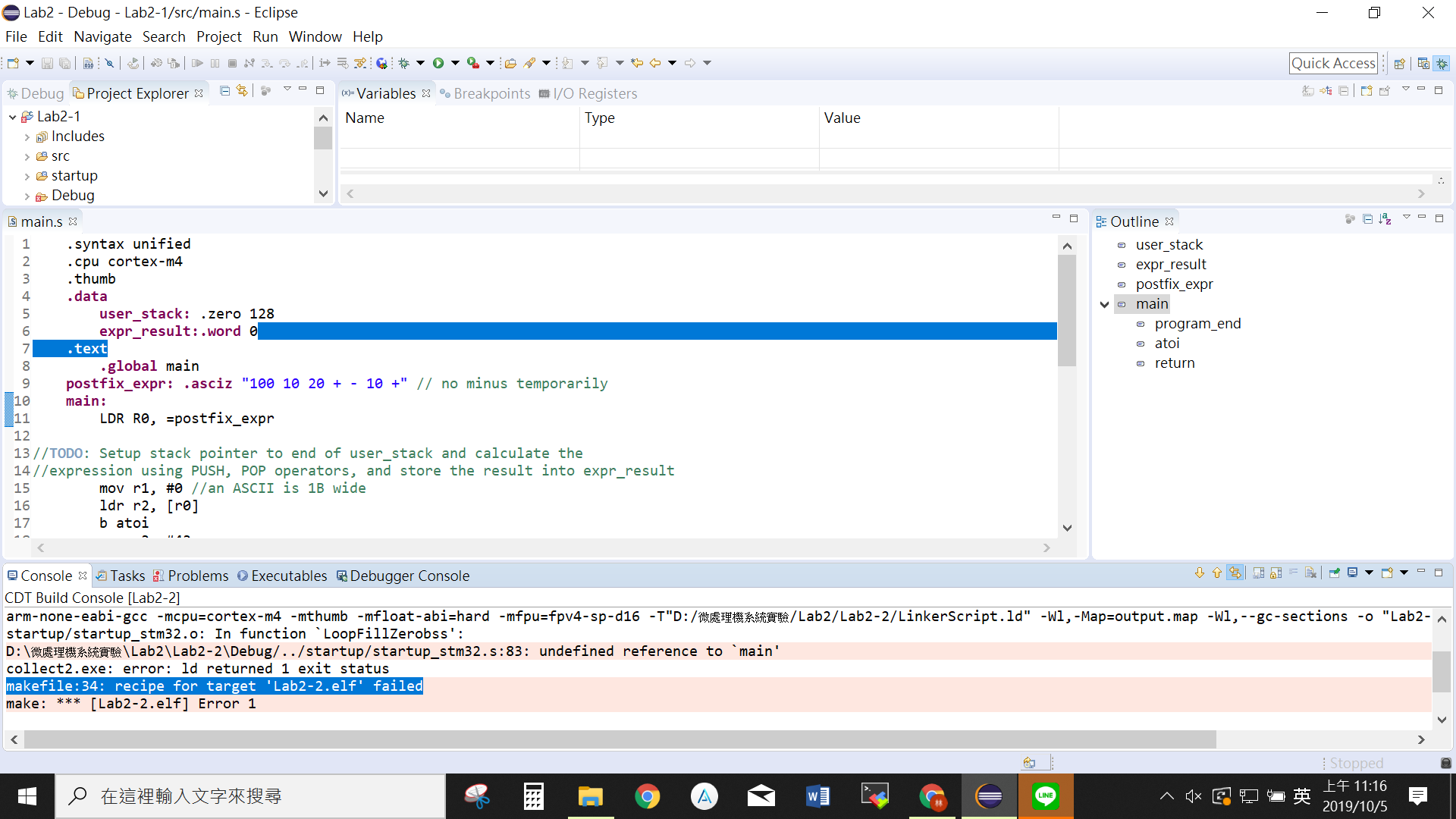
回main後，先將最後一次取出的乘數與r6相乘，得到正確的結果。再將

記錄stack size的r8乘上8已表示byte。最後再將此兩項存入result

與max\_size的位置即可完成。

1. **Feedback 實驗心得~~或~~建議** 

**Lab2 PDF內，把ascii放在text section(程式文字檔區)會被誤認為是instruction.**



**Solution on StackOverflow:**

