微處理機實驗報告 LAB3

0416094 黃兆宇

0416033 黃品任

1. 實驗名稱

ARM assembly 2

1. 實驗目的

1.藉由實作postfix的運算熟悉arm組合語言中atoi的方式和stack的使用

2.藉由實作stein算法來熟悉arm組合語言中的recursion用法和stack使用

1. 實驗步驟

第一題 postfix

一次讀入一個byte並以r3標記此byte的屬性(r3 = 3 加號、r3 = 2 減號

、r3 = 0 正數 、 r3 = 1 負數)為區別減號和負數，故先判斷加減號和是否後方有數字，減號後方有數字即為負數。

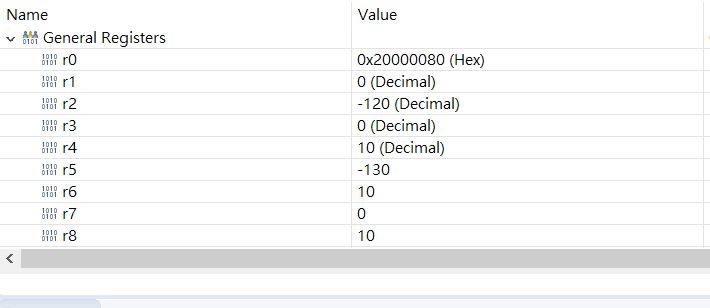
接著進行stack的運算，先將數字依次推入stack中，並一次推出兩個數字進行加減運算，判斷出’/0’字元即跳出。

第二題 stein 算 gcd

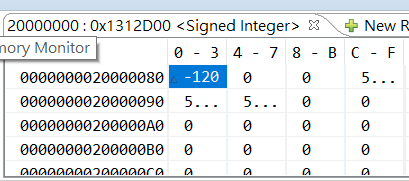
依照stein演算法算gcd，用recursion的方式，將lr , m , n值傳入下一個gcd function 中，在下一個function中pop m , n的值，使得stack中只剩下lr，執行到中止條件後，在依次pop出之前存入stack中的lr，並依序跳回，直到回到起始的function，過程中也記錄乘2的次數(r6)，並在最後將結果向左shift r6 次，得到答案。

1. 實驗結果與分析

第一題:

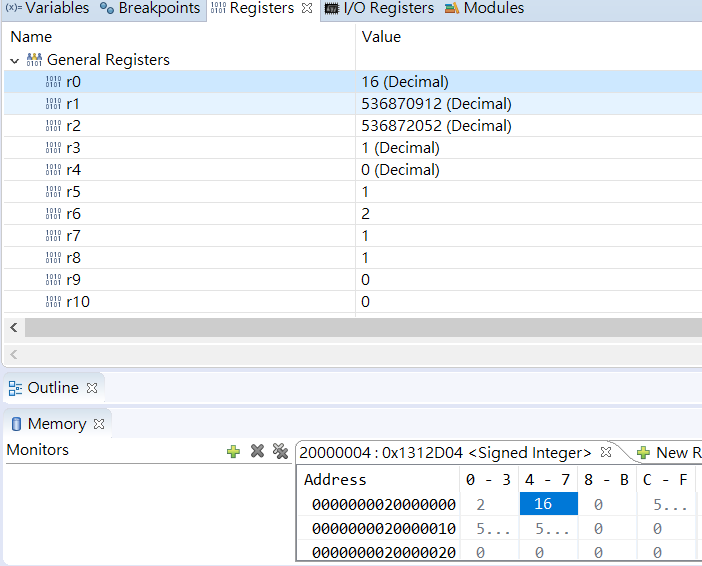


R2 是答案 -120



Expr\_result = 120

第二題 :



結果 result 是 2 (94,96的公因數) max\_size 是 16

1. 心得討論與應用聯想

這次的兩次題目都跟stack有關，第一題的字串轉換因為要處理負號的關係，變得比較複雜，要區分負號和減號，必須使用flag判斷。

而第二體主要是要處理許多判斷式和函數的參數傳遞問題，原本在遞迴之後，想直接跳回主程式，但跳躍的距離太遠了，會進入無窮迴圈，最後只好以stack儲存，依次跳回。