

## **實驗名稱：實驗二 ARM Assembly I**

### **實驗目的：**

熟悉基本 ARMv7 組合語言語法使用。

並了解以下幾點：

- 1.如何利用條件跳躍指令完成程式迴圈的操作
- 2.算數與邏輯操作指令使用
- 3.暫存器(Register)使用與基本函式參數傳遞
- 4.記憶體與陣列存取

### **實驗步驟：**

#### **1. Hamming distance**

計算兩個數長度為 half-word(2bytes)的漢明距離，並將結果存放至 result 變數中。

#### **2. Fibonacci serial**

宣告一數值 N ( $1 \leq N \leq 100$ )，計算 Fib(N)並將回傳值存放至 R4 暫存器。

#### **3. Bubble sort**

利用組合語言完成長度為 8byte 的 8bit 泡沫排序法。完成 do\_sort 函式，其中陣列起始記憶體位置作為輸入參數 R0，程式結束後需觀察 arr1 與 arr2 記憶體內容是否有排序完成。

## 實驗結果：

### 1. Hamming distance

```
5 .data
6     result: .byte 0
7 .text
8 //global main
9     .equ X, 0x55AA
10    .equ Y, 0xAA55
11 hamm:
12    EOR r0, r1
13    lsl r0, #16 //shift to high bits
14    movs r3, #16 //loop counter
15    movs r4, #0 //result
16 loop:
17    lsl r0, #1
18    adc r4, #0
19    subs r3, #1
20    bne loop
21    str r4, [r2]
22    bx lr
23 main:
24    ldr R0, =X
25    ldr R1, =Y
26    ldr R2, =result
27    bl hamm
28 L: b L
```

其中line 24與line 25不能使用movs指令將X,Y的值存入register 內，因為X、Y的值超過movs 指令的範圍，因此需用ldr。

### 2. Fibonacci serial

```
6 .text
7     .global main
8     .equ N, 20
9
10 fib:
11     movs r4, #-1
12     CMP r0, 100
13     BLE continue1 //N<=100
14     bx lr
15 continue1:
16     CMP r0, #0
17     BGT continue2 //N>=1
18     bx lr
19 continue2:
20     movs r4, #1 // fib(1)
21     movs r3, #0 // fib(0)
22 loop:
23     CMP r0, #1
24     BEQ exit
25     adds r5, r3, r4 // fib(n) = fib(n-1) + fib(n-2)
26     BVS of //overflow
27     movs r3, r4
28     movs r4, r5
29     subs r0, #1
30     B loop
31 exit:
32     bx lr
33 of:
34     mov r4, #-2 //return -2
35     bx lr
36 main:
37     movs R0, #N
38     bl fib
39 L: b L
```

### 3. Bubble sort

以下為 do\_sort 函式:

```
11 do_sort:
12     movs r4, #7 //outer loop counter
13 outer_loop:
14     movs r5, #0 //inner counter counter
15
16 inner_loop:
17     movs r6, r5
18     adds r6, #1
19     ldrb r1, [r0,r5]
20     ldrb r2, [r0,r6]
21     cmp r1, r2
22     bls no_swap //less than
23     strb r1, [r0,r6] //swap
24     strb r2, [r0,r5] //swap
25 no_swap:
26     adds r5, #1
27     cmp r5, r4
28     bne inner_loop
29     subs r4, #1
30     BNE outer_loop
31     bx lr;
```

#### 心得討論與應用聯想:

此次實驗主要是讓我們練習利用 assembly language 來實作條件判斷、迴圈的操作，以及簡單的函式呼叫。基本上所有的程式都是以這些條件判斷與迴圈所建構，因此能熟練運用此次實驗的技巧就能實作出大部分我們所需的程式。