



實驗一 實驗環境建立與Debugger操作

0516039 羅文笙

1. Lab objectives 實驗目的

- (1) Test the kit.
- (2) Familiar with the development environment.

2. Steps 實驗步驟

2.1 專案建立與程式編譯

- step 1:根據lab0的步驟建立c project
- step 2:打上老師附的程式碼
- step 3:按下鐵鎚的按鈕來編譯程式
- step 4:按下debug，並持續按f5來觀察register view內各個register值的變化

2.2 變數宣告與記憶體觀察

- Q1:觀察程式碼即可
- Q2:將.data內的資料剪下貼到.text下面即可
- Q3:按下debug後，持續按f5直到最後一行，接著觀察register和memory的值

2.3 簡易算數與基本記憶體指令操作

- step 1:打出程式碼



```
1      .syntax unified
2      .cpu cortex-m4
3      .thumb
4
5  .data
6      X: .word 5
7      Y: .word 10
8      Z: .word
9
10 .text
11 .global main
12
13 main:
14     ldr    r0, =X//r0 = X's address
15     ldr    r1, =Y//r1 = Y's address
16     ldr    r2, =Z//r2 = Z's address
17     ldr    r3, [r0]//r3=X=5
18     ldr    r4, [r1]//r4=Y=10
19     mul    r5,r3,r4//r5=x*10=x*y
20     adds   r5,r5,r4//r5=x*10+y
21     subs   r6,r4,r3//r6=Y-X
22     str    r5,[r0]//x=r5
23     str    r6,[r2]//z=r6
24 L: B L
25
```

step 2:觀察register和 memory即可

3. Results and analysis 實驗結果與分析

2.1

Q1:R2的值為105

(x)= Variables Breakpoints 1010 0101 Registers 23 I/O Registers Modules	
Name	Value
General Registers	
1010 0101 r0	85
1010 0101 r1	20
1010 0101 r2	105



2.2

Q1:在.data的部分

```
.data
X: .word 100
str: .asciz "Hello World!"
```

Q2:會出現編譯錯誤

```
.. gcc (SystemWorkbench) test (Debug)
arm-none-eabi-as -mcpu=cortex-m4 -mthumb -mfloat-abi=hard -mfpu=fpv4-sp
../src/main.s: Assembler messages:
../src/main.s: Error: unaligned opcodes detected in executable segment
make: *** [src/main.o] Error 1

16:41:38 Build Finished (took 187ms)
```

Q3: 似乎是反過來的，r2內的值是6C6C6548(十進位是1819043144)，
而str的前4 bytes是48656C6C (48代表H, 65代表e, 6C代表l)

Q4: **str: .byte 'H', 'e', 'l', 'l', 'o', ' ', 'W', 'o', 'r', 'l', 'd', '!'**

2.3

Q1: Address of X,Y,Z: 0x20000000, 0x20000004, 0x20000008

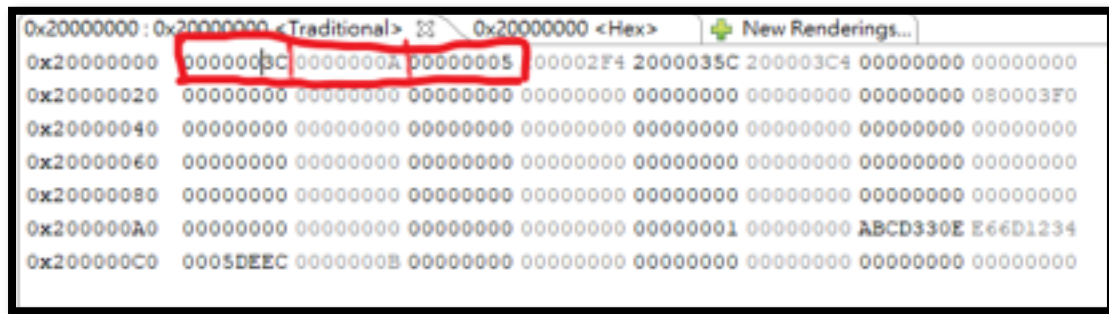
(r0~r2的值換成16進位就會變成0x20000000, 0x20000004, 0x20000008)

(r3 = X的初始值，r4 = Y的初始值，r5 = X最後的值，r6 = Z的值)

x)= Variables Breakpoints Registers I/O Registers Modules	
Name	Value
General Registers	
r0	536870912
r1	536870916
r2	536870920
r3	5
r4	10
r5	60
r6	5



Memory table:可以看出最後X, Y, Z內的值分別為 60, 10, 5



0x20000000	0x20000000	<Traditional>	32	0x20000000	<Hex>	New Renderings...
0x20000000	0000003C	0000000A	00000005	00002F4	2000033C	200003C4 00000000 00000000
0x20000020	00000000	00000000	00000000	00000000	00000000	00000000 080003F0
0x20000040	00000000	00000000	00000000	00000000	00000000	00000000 00000000
0x20000060	00000000	00000000	00000000	00000000	00000000	00000000 00000000
0x20000080	00000000	00000000	00000000	00000000	00000000	00000000 00000000
0x200000A0	00000000	00000000	00000000	00000000	00000001	00000000 ABCD330E E66D1234
0x200000C0	0005DEEC	0000000B	00000000	00000000	00000000	00000000 00000000

4. Conclusions and ideas 心得討論與應用聯想

從這一次的實驗中，我對於這個ide和arm 有了初步的認識，在每一次的錯誤和嘗試後，我都對於這個環境有更深一層的了解，我想我已經準備好接下來更難的實驗了！