

# 第04次組語實習課

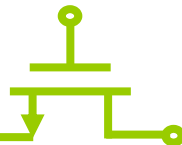
學生：林培瑋

2023 Advanced Mixed-Operation System (AMOS) Lab.



**Tamkang University**  
**Department of Electrical and Computer Engineering**  
No.151, Yingzhuan Rd., Tamsui Dist., New Taipei City 25137, Taiwan (R.O.C.)





❖ 第1次隨堂考.....	3
❖ 第一次作業.....	5
❖ 第二次作業.....	14



# 第1次隨堂考

**2023 Advanced Mixed-Operation System (AMOS) Lab.**



**Tamkang University**  
**Department of Electrical and Computer Engineering**  
No.151, Yingzhuan Rd., Tamsui Dist., New Taipei City 25137, Taiwan (R.O.C.)





0	沒繳交、交白券
20	基本分(有繳交計算過程、只交程式碼、沒進入Debugger介面)
40	有進入Debugger介面，程式碼是錯的
60	程式碼有小錯誤，導致輸出結果數值不正確
80	沒寫計算過程、計算過程不完整
100	全對

➤ 下次隨堂考開始，將不會有部分給分。



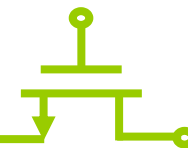
# 第一次作業

**2023 Advanced Mixed-Operation System (AMOS) Lab.**



**Tamkang University**  
**Department of Electrical and Computer Engineering**  
No.151, Yingzhuan Rd., Tamsui Dist., New Taipei City 25137, Taiwan (R.O.C.)





- ❖ 第一部分(30%) 至少2張截圖(一張15%) (**執行一次LSL**一張截圖)。
- ❖ 第二部分(30%)至少2張截圖(一張15%) (**執行第一次迴圈結束時**一張截圖，程式**執行結束**一張截圖)。
- ❖ 第三部分(30%)至少3張截圖(一張10%) (**執行一次EOR**一張截圖)。
- ❖ 心得(10%)

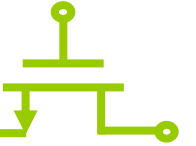
不計分	一半分數
程式碼沒有學號、姓名	沒截到記憶體視窗
暫存器模糊不清楚	更改範例程式的程式碼
暫存器視窗沒拉開	暫存器視窗拉太寬
暫存器數值不正確	螞蟻圖(下次作業開始)

➤ 以最後繳交的版本為準

➤ 遲交者成績 = 原始成績\*0.5



# 第一部份(執行一次LSL)



Registers

Register	Value
<b>Current</b>	
R0	0x00000011
R1	0x00000022
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000008
CPSR	0x000000D3
SPSR	0x00000000
<b>User/System</b>	
Fast Interrupt	
Interrupt	
<b>Supervisor</b>	
Abort	
Undefined	
<b>Internal</b>	
PC \$	0x00000008
Mode	Supervisor
States	2
Sec	0.00000017

Disassembly

```

0x00000000 E3A00011 MOV R0,#0x00000011
5:          LSL r1, r0, #1
0x00000004 E1A01080 MOV R1,R0,LSL #1
6:          LSL r2, r1, #1
7:
->0x00000008 E1A02081 MOV R2,R1,LSL #1

```

HW1.1.s

```

1          AREA   prog1, CODE, READONLY
2          ENTRY
3
4          MOV    r0, #0x11
5          LSL    r1, r0, #1
6          LSL    r2, r1, #1
7
8          stop   B      stop
9
10         END

```

Memory 1

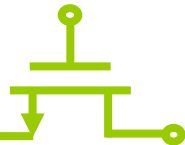
Address: 0x00

0x00000000:	11 00 A0 E3 80 10 A0 E1 81 20 A0 E1 FE FF FF EA 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0000001B:	00 00
0x00000036:	00 00

Call Stack + Locals

Memory 1

# 第一部份(執行一次LSL)



Registers

Register	Value
Current	
R0	0x00000011
R1	0x00000022
R2	0x00000044
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x0000000C
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x0000000C
Mode	Supervisor
States	3
Sec	0.00000025

Disassembly

```

0x00000008 E1A02081 MOV R2,R1,LSL #1
8: stop B stop
0x0000000C EAffffff B 0x0000000C
0x00000010 00000000 ANDEQ R0,R0,R0
0x00000014 00000000 ANDEQ R0,R0,R0
0x00000018 00000000 ANDEQ R0,R0,R0
0x0000001C 00000000 ANDEQ R0,R0,R0

```

HW1.1.s

```

1 AREA prog1, CODE, READONLY
2 ENTRY
3
4 MOV r0, #0x11
5 LSL r1, r0, #1
6 LSL r2, r1, #1
7
8 stop B stop
9
10 END

```

Memory 1

Address: 0x00

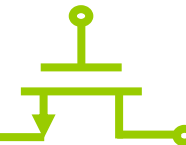
0x00000000:	11 00 A0 E3 80 10 A0 E1 81 20 A0 E1 FE FF FF EA 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x0000001B:	00 00
0x00000036:	00 00

Call Stack + Locals

Memory 1



## 第二部份(執行第一次迴圈結束時)



Registers

Register	Value
<b>Current</b>	
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000009
R7	0x0000000A
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
<b>R15 (PC)</b>	<b>0x00000008</b>
CPSR	0x200000D3
SPSR	0x00000000
<b>User/System</b>	
Fast Interrupt	
Interrupt	
<b>Supervisor</b>	
Abort	
<b>Undefined</b>	
<b>Internal</b>	
PC \$	0x00000008
Mode	Supervisor
States	9
Sec	0.00000075

Disassembly

```

0x00000004 E3A07001 MOV R7,#0x00000001
6: loop CMP r6,#0
0x00000008 E3560000 CMP R6,#0x00000000
7: MULGT r7,r6,r7
0x0000000C C0070796 MULGT R7,R6,R7
8: SUBGT r6,r6,#1
0x00000010 E3460001 SUBGT R6,R6,#0x00000001

```

HW1.2.s

```

1 AREA prog2, CODE, READONLY
2 ENTRY
3
4 MOV r6, #10
5 MOV r7, #1
6 loop CMP r6, #0
7 MULGT r7, r6, r7
8 SUBGT r6, r6, #1
9 BGT loop
10
11 stop B stop
12 END

```

Memory 1

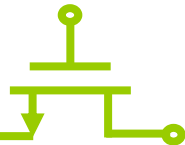
Address: 0x00000000

0x00000000:	0A 60 A0 E3 01 70 A0 E3 00 00 56 E3 96 07 07 C0 01 60 46 C2 FB FF FF CA FE
0x00000019:	FF FF EA 00
0x00000032:	00 00
0x0000004B:	00 00

Call Stack + Locals

Memory 1

## 第二部份(執行結束)



Registers

Register	Value
<b>Current</b>	
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00375F00
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000018
CPSR	0x600000D3
SPSR	0x00000000
<b>User/System</b>	
Fast Interrupt	
Interrupt	
<b>Supervisor</b>	
Abort	
Undefined	
<b>Internal</b>	
PC \$	0x00000018
Mode	Supervisor
States	123
Sec	0.00001025

Disassembly

```

11: stop    B      stop
0x00000018 EAffffff B      0x00000018
0x0000001C 00000000 ANDEQ   R0,R0,R0
0x00000020 00000000 ANDEQ   R0,R0,R0
0x00000024 00000000 ANDEQ   R0,R0,R0
0x00000028 00000000 ANDEQ   R0,R0,R0
0x0000002C 00000000 ANDEQ   R0,R0,R0

```

HW1.2.s

```

1      AREA    prog2, CODE, READONLY
2      ENTRY
3
4      MOV     r6, #10
5      MOV     r7, #1
6  loop  CMP     r6, #0
7      MULGT   r7, r6, r7
8      SUBGT   r6, r6, #1
9      BGT     loop
10
11  stop  B      stop
12      END

```

Memory 1

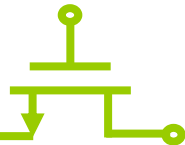
Address: 0x00000000

0x00000000:	0A 60 A0 E3 01 70 A0 E3 00 00 56 E3 96 07 07 C0 01 60 46 C2 FB FF FF CA FE
0x00000019:	FF FF EA 00
0x00000032:	00 00
0x0000004B:	00 00

Call Stack + Locals

Memory 1

## 第三部份(執行一次EOR)



Registers

Register	Value
<b>Current</b>	
R0	0xE16298F1
R1	0x17539ABD
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
<b>R15 (PC)</b>	<b>0x0000000C</b>
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
<b>Supervisor</b>	
Abort	
Undefined	
Internal	
PC \$	0x0000000C
Mode	Supervisor
States	7
Sec	0.00000058

Disassembly

```

7:          EOR      r1, r0, r1
0x0000000C E0201001 EOR      R1,R0,R1
8:          EOR      r0, r0, r1
9:
0x00000010 E0200001 EOR      R0,R0,R1
10: stop      B      stop
0x00000014 F1FFFFFF B      0x00000014

```

HW1.3.s

```

1          AREA      prog3, CODE, READONLY
2          ENTRY
3
4          LDR      r0, =0xF631024C
5          LDR      r1, =0x17539ABD
6
7          EOR      r0, r0, r1
8          EOR      r1, r0, r1
9
10         stop      B      stop
11         END

```

Memory 1

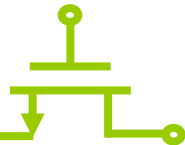
Address: 0x00000000

0x00000000:	10 00 9F E5 10 10 9F E5 01 00 20 E0 01 10 20 E0 01 00 20 E0 FE FF FF EA
0x00000018:	4C 02 31 F6 BD 9A 53 17 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000030:	00 00
0x00000048:	00 00

Call Stack + Locals

Memory 1

## 第三部份(執行一次EOR)



Register	Value
<b>Current</b>	
R0	0xE16298F1
R1	0xF631024C
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000010
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
<b>Supervisor</b>	
Abort	
Undefined	
Internal	
PC \$	0x00000010
Mode	Supervisor
States	8
Sec	0.00000067

Disassembly

```

7:          EOR      r1, r0, r1
0x0000000C E0201001 EOR      R1,R0,R1
8:          EOR      r0, r0, r1
9:
0x00000010 E0200001 EOR      R0,R0,R1
10: stop      B      stop
0x00000014 F1FFFFFF B      0x00000014

```

HW1.3.s

```

1          AREA      prog3, CODE, READONLY
2          ENTRY
3
4          LDR      r0, =0xF631024C
5          LDR      r1, =0x17539ABD
6          EOR      r0, r0, r1
7          EOR      r1, r0, r1
8          EOR      r0, r0, r1
9
10         stop      B      stop
11         END

```

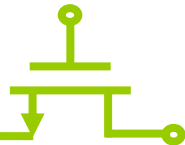
Memory 1

Address: 0x00000000

0x00000000:	10 00 9F E5 10 10 9F E5 01 00 20 E0 01 10 20 E0 01 00 20 E0 FE FF FF EA
0x00000018:	4C 02 31 F6 BD 9A 53 17 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000030:	00 00
0x00000048:	00 00

Call Stack + Locals    Memory 1

## 第三部份(執行一次EOR)



Register	Value
<b>Current</b>	
R0	0x17539ABD
R1	0xF631024C
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000014
CPSR	0x000000D3
SPSR	0x00000000
<b>User/System</b>	
Fast Interrupt	
Interrupt	
<b>Supervisor</b>	
Abort	
Undefined	
<b>Internal</b>	
PC \$	0x00000014
Mode	Supervisor
States	9
Sec	0.00000075

Disassembly

```

10: stop    B      stop
0x00000014 EAfffffe B      0x00000014
0x00000018 F631024C (???)
0x0000001C 17539ABD ???NE
0x00000020 00000000 ANDEQ    R0,R0,R0
0x00000024 00000000 ANDEQ    R0,R0,R0
0x00000028 00000000 ANDEQ    R0,R0,R0

```

HW1.3.s

```

1      AREA    prog3, CODE, READONLY
2
3      ENTRY
4
5      LDR      r0, =0xF631024C
6      LDR      r1, =0x17539ABD
7      EOR      r0, r0, r1
8      EOR      r1, r0, r1
9      EOR      r0, r0, r1
10     stop    B      stop
11     END

```

Memory 1

Address: 0x00000000

0x00000000:	10 00 9F E5 10 10 9F E5 01 00 20 E0 01 10 20 E0 01 00 20 E0 FE FF FF EA
0x00000018:	4C 02 31 F6 BD 9A 53 17 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x00000030:	00 00
0x00000048:	00 00

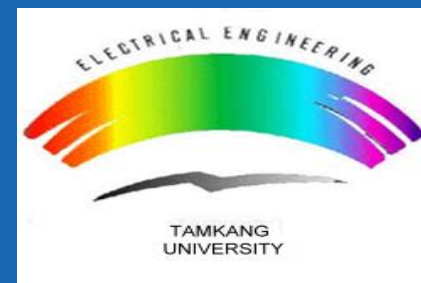
Call Stack + Locals    Memory 1

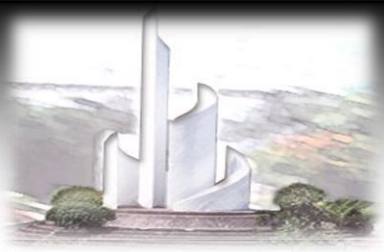
# 第二次作業

**2023 Advanced Mixed-Operation System (AMOS) Lab.**



**Tamkang University**  
**Department of Electrical and Computer Engineering**  
No.151, Yingzhuan Rd., Tamsui Dist., New Taipei City 25137, Taiwan (R.O.C.)





# 組合語言\_HW2

繳交期限: 11/2 23:59

(用word檔上傳至iclass)

請提前上傳，以免iclass塞車

**PANEL**

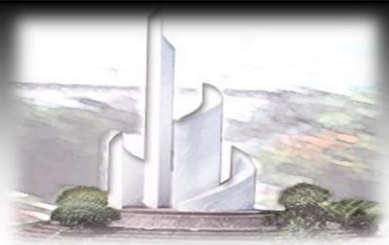
Parallel Architectures & Networks Lab.

TAMU of Electrical Engineering 淡江大學電機系





# 第一部分



1. 照著課本P.57第6題的要求，寫出一個程式

➤ 求出  $6x^2 - 9x + 2$  的結果值

➤  $R3 = x$

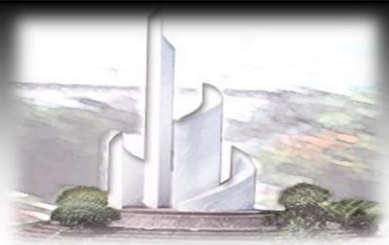
➤  $R2 = \text{答案}$

➤  $R3$  的  $x$  請設成 731 (0x2DB)

➤ 最後程式跑出來， $R2$  的值要跟自己設的數代進去方程式算出來一樣。



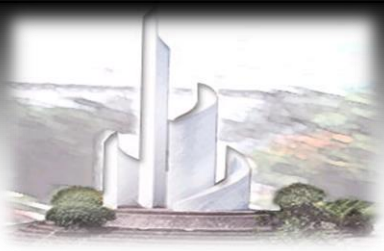
# 第一部分



## 2. 課本p.58第8題

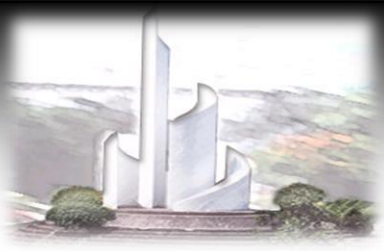
- 利用two's complement用手算-149和-4321兩個值轉成16進位，如同program 3用LDR放進暫存器R0和R1
- 把兩個值加起來存到暫存器R7
- 將R7暫存器的16進位數值，用手算逆推回十進制，驗證程式計算結果是否正確

# 第一部分



- ▶ 將前面兩大題相關程式如上課時所示範的一樣，各打入一個program，並單步執行（F11），且截圖展示出題目所要求的結果，手算驗證部分用註解呈現在program中

## 第二部分



data1 DCW 0x8ECC, 0xFE37, -149  
data2 DCD 0xFE37, 1, 5, 20  
data3 DCB 0xCF, 23, 39, 0x54, 250  
data4 DCWU 0x1234  
data5 DCB 255  
data6 DCDU 0x12345678, -4321  
data7 DCB 0xA3  
ALIGN 4,3  
data8 DCWU 0xFC25  
ALIGN  
data9 DCB “MVP\_N. Jokic”, 0  
data10 DCW 0xEF12

**PANEL**

Parallel Architectures & Networks Lab.

TJMU of Electrical Engineering 淡江大學電機系

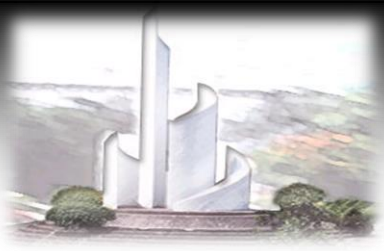


## 第二部分



- ▶ 如上課時所示範的一樣，將前一頁之宣稱寫入程式
- ▶ 在記憶體視窗中紅色標出每一個值儲存之位置並註明 data1~data10 每一變數的位址

# 第三部分

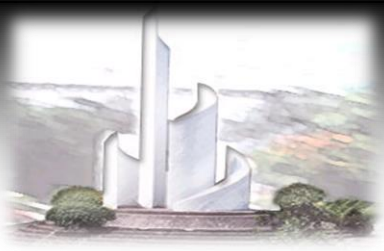


- 1. (1) Assume  $r3 = 0x40000000$  and  $r4 = 0x60$ . What would  $r3$  contain
- after executing each of the following instructions?
- (a) `STR r6, [r3, #8]`
- (b) `STRB r7, [r3], #12`
- (c) `LDRH r5, [r3], #12`
- (d) `LDR r12, [r3, #4]!`
- (e) `LDR r6, [r3, r4, ROR #28]!`
- (f) `LDR r0, [r3, r4, LSL #2]`
- (2) What would register  $r4$  contain after executing the following
- instructions? Register  $r6$  holds the value `0xDEADBEEF`
- and register  $r3$  holds `0x40000000`.
- `STR r6, [r3]`
- `LDRB r4, [r3]`





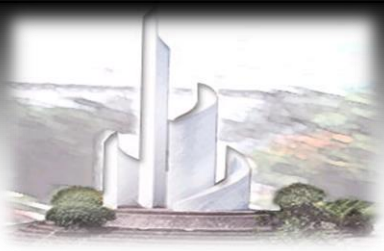
# 第三部分



- 2. If  $r2 = 0x12345678$ ,  $r3 = 0x87654321$ , and  $r4 = 0x00000012$  before execution, give the contents of  $r2$  after each of the following instructions is executed.
- (1) BIC  $r2, r2, \#0xFF000000$
- (2) LSL  $r2, r3, \#4$
- (3) LSL  $r2, r2, r4$
- (4) ROR  $r2, r2, \#12$  ;
- (5) AND  $r2, r2, r3$
- (6) ORR  $r2, r2, r4$
- (7) EOR  $r2, r2, r4$
- (8) BIC  $r2, r2, r4$
- (9) EOR  $r2, r2, r3$ , ROR  $\#7$



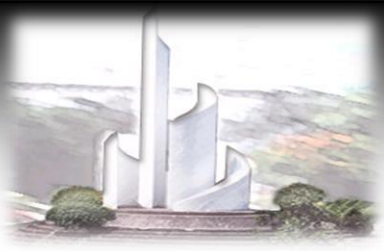
# 第三部分



- 3. Assume  $r6 = 0xABCD8765$ . Write a sequence of instructions to
  - (1) calculate the 2's complement of  $r6$  and put the result in  $r7$ .
  - (2) set bits 1, 5, and 13 in register  $r6$  and leave the remaining bits unchanged.
  - (3) clear bits 0, 4, and 12 in register  $r6$  and leave the remaining bits unchanged.
  - (4) change bits 4, 8, and 11 of  $r6$ .
  - (5) insert the value  $0x5555$  into the lower half of register  $r0$  so that the final value is  $0xBEEF5555$ , assuming register  $r0$  contains the value  $0xBEEFABCD$ .



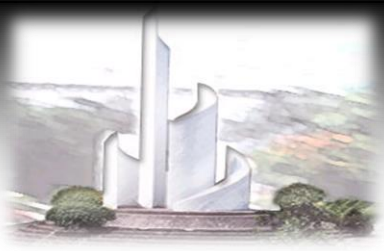
# 第三部分



- ▶ 將前面三大題相關程式如上課時所示範的一樣，各打入一個program，並單步執行（F11），且截圖展示出題目所要求的變化







- ◆ 繳交內容：按照結報格式寫完整，把結報word上傳iclass對應作業位置。
- ◆ 程式**需在Keil Tool程式視窗內展示**且展示程式與執行結果的**每一截圖需看的到學號姓名**否則不計分。
- ◆ 繳交期限：**11/2 23:59**
- ◆ Word檔名：**組合語言\_學號\_姓名\_HW2**

# *Q&A*

***Thanks for your attention !!***