第10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.)

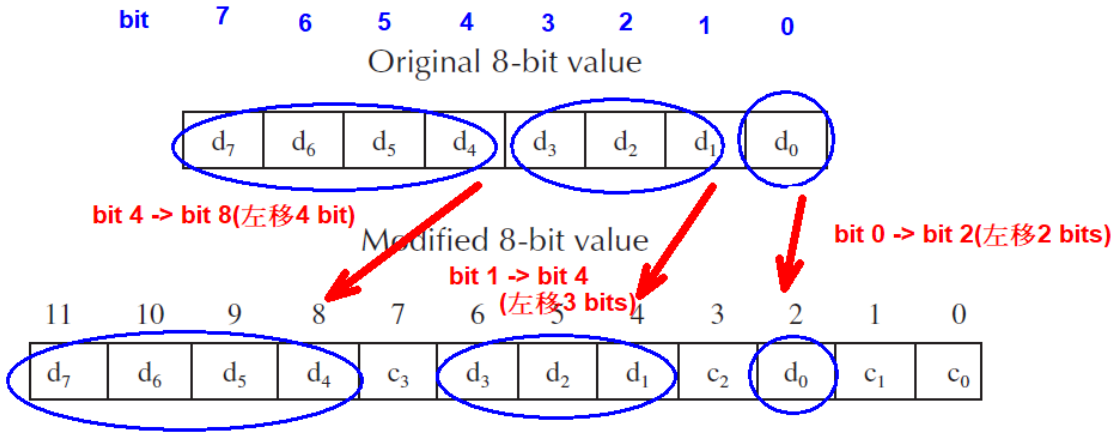
目錄

1121正課複習

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.)

p.132 Build The Final 12-bit Result



132 ARM Assembly Language

```
EOR r4, r4, r0, ROR #6      ; 6 XOR 5 XOR 4
EOR r4, r4, r0, ROR #7      ; 7 XOR 6 XOR 5 XOR 4
AND r4, r4, #1
;
; build the final 12-bit result
;
ORR r2, r2, r4, ROR #25     ; rotate left 7 bits
AND r4, r0, #1              ; get bit 0 from original
ORR r2, r2, r4, LSL #2      ; add bit 0 into final
BIC r4, r0, #0xF1           ; get bits 3,2,1
ORR r2, r2, r4, LSL #3      ; add bits 3,2,1 to final
BIC r4, r0, #0x0F           ; get upper nibble
ORR r2, r2, r4, LSL #4      ; r2 now contains 12 bits
                            ; with checksums
```

可以寫成
LDR r5, =0xFFFFFFFF
BIC r4, r0, r5

建議寫成
AND r4, r0, #0xE

AND r4, r0, #0xF0

因為已知前3個Bytes為0

Hamming Code補充題型(1/4)

1. Get received checksum bits c_0, c_1, c_2, c_3 from the received 12 bits and respectively store in R6, R7, R8, R9.

   (1) Get bit 0 from the received 12 bits

   (Store 12 bits into R0→R0 = 0 or 1→ c_0 = 0 or 1)

   (2) Get bit 1 from the received 12 bits

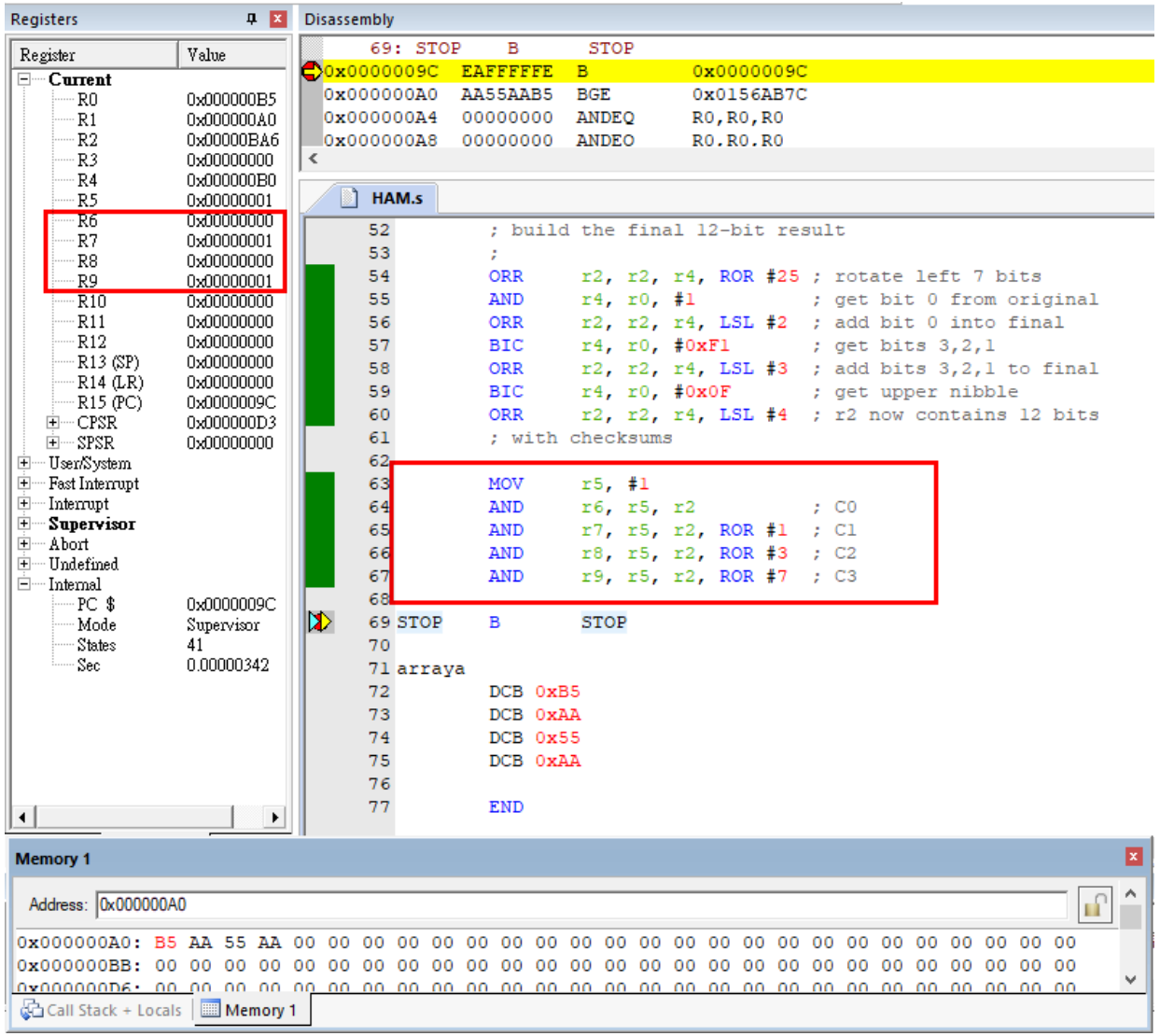
   (3) Get bit 3 from the received 12 bits

   (4) Get bit 7 from the received 12 bits

Hamming Code補充題型(1/4)

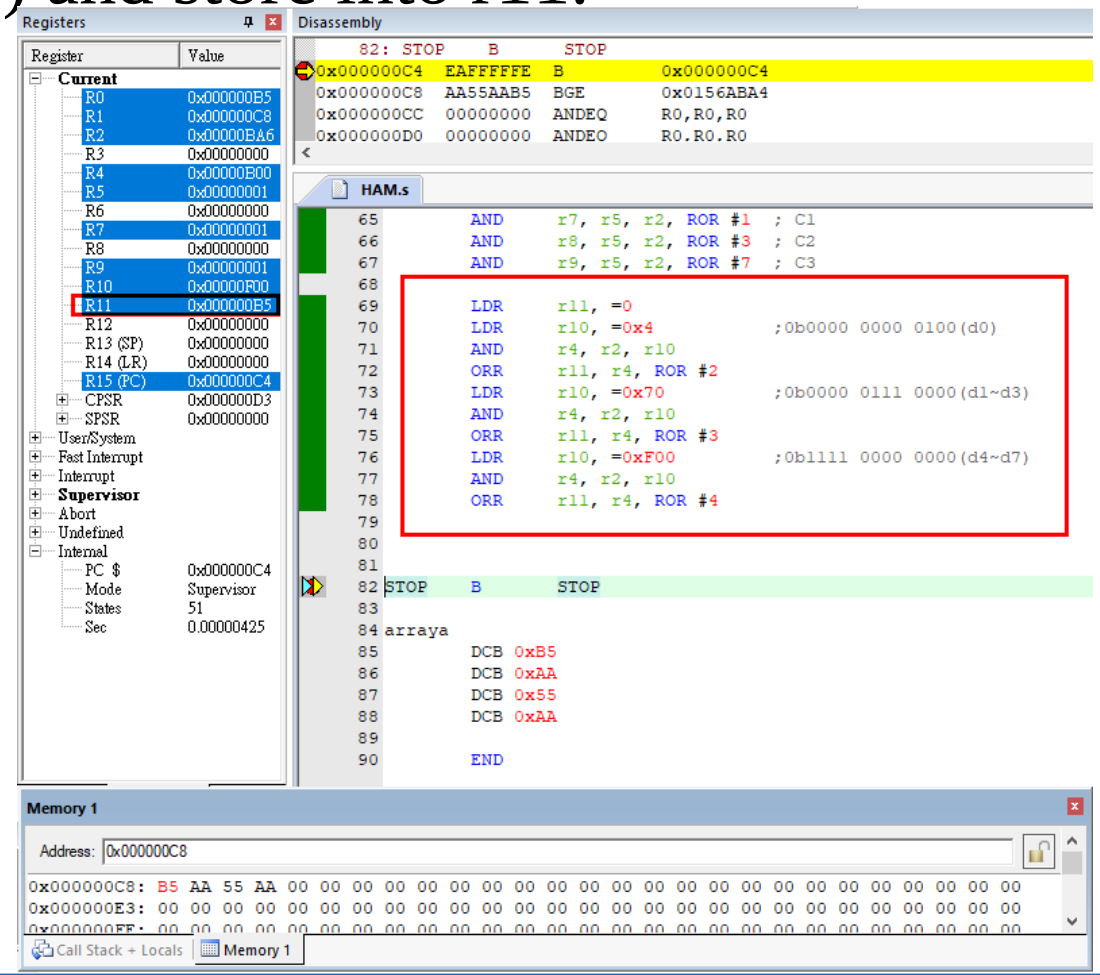
→Original 8-bit value

→Modified 8-bit value



Hamming Code補充題型(2/4)

2. Get the received data bits from the received 12 bits (get d7~d0 from the received 12 bits) and store into r11.



Hamming Code補充題型(3/4)

3. Compute checksum bits c0, c1, c2, c3 (computed checksum bits c0, c1, c2, c3) from the received data bits.

Hamming Code補充題型(4/4)

4. Compare received with computed c0, c1, c2, c3.
   1) No bit different(correct data bits received)
   2) One bits different(checksum bit itself incorrect and data bits correct)
   3) Two bits different(one of received data bits is incorrect d_j) →$j = (2^n + 2^m) - 1$
   4) Three bits different(暫時不討論)
   5) Four bits different(暫時不討論)

第3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.)

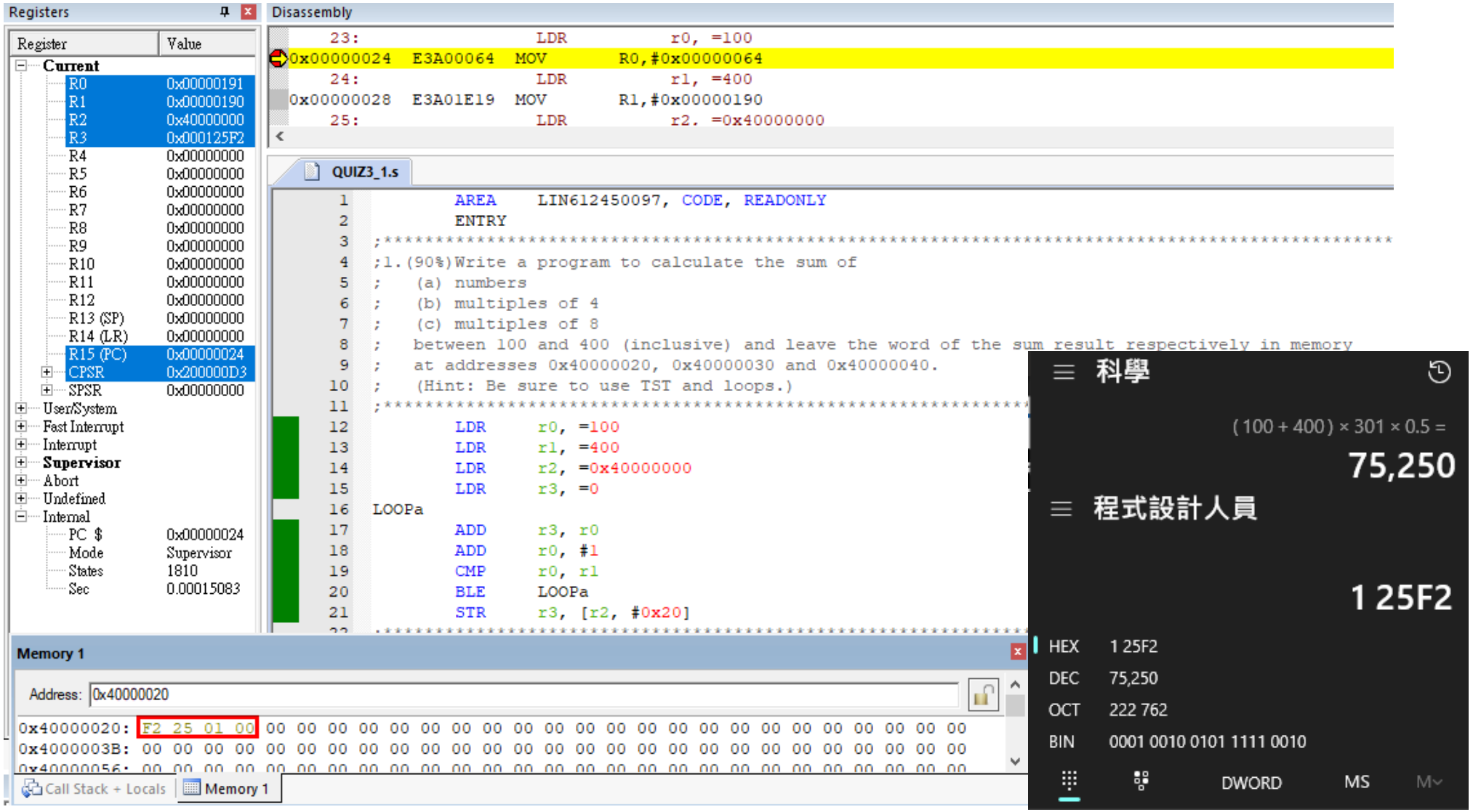
評分標準

→共有3題，一題30分。(若符合第二題規定+10分)

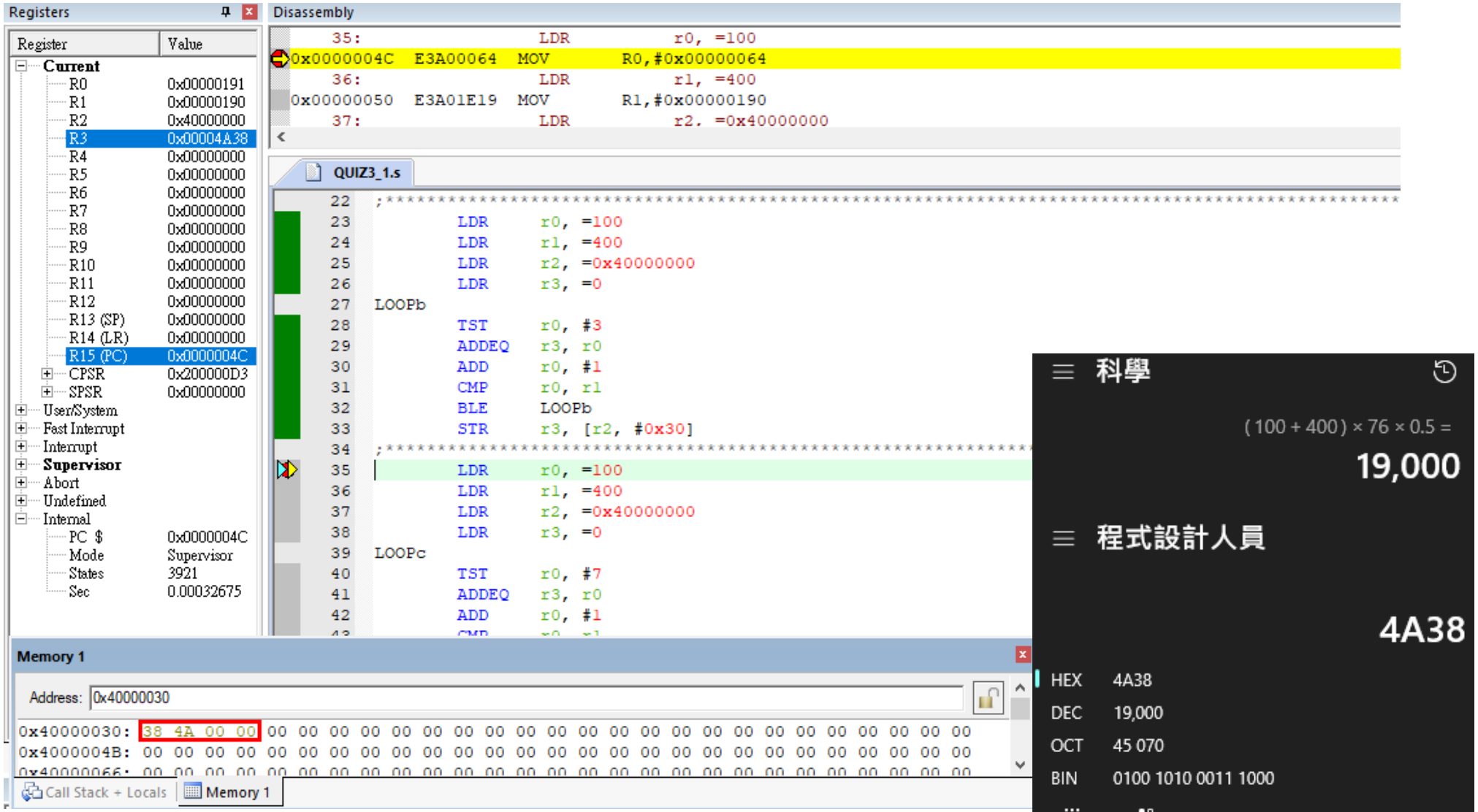
0	未繳交、交白券、**程式碼裡沒學號姓名**
5	基本分(只交程式碼，沒進入Debugger介面)
10	有進入Debugger介面，程式碼與題目要求的差很多
20	程式碼有小錯誤，導致輸出結果數值不正確
25	輸出結果數值正確，但未附上小算盤驗算截圖
30	完全正確

→補繳分數=原始分數*0.9

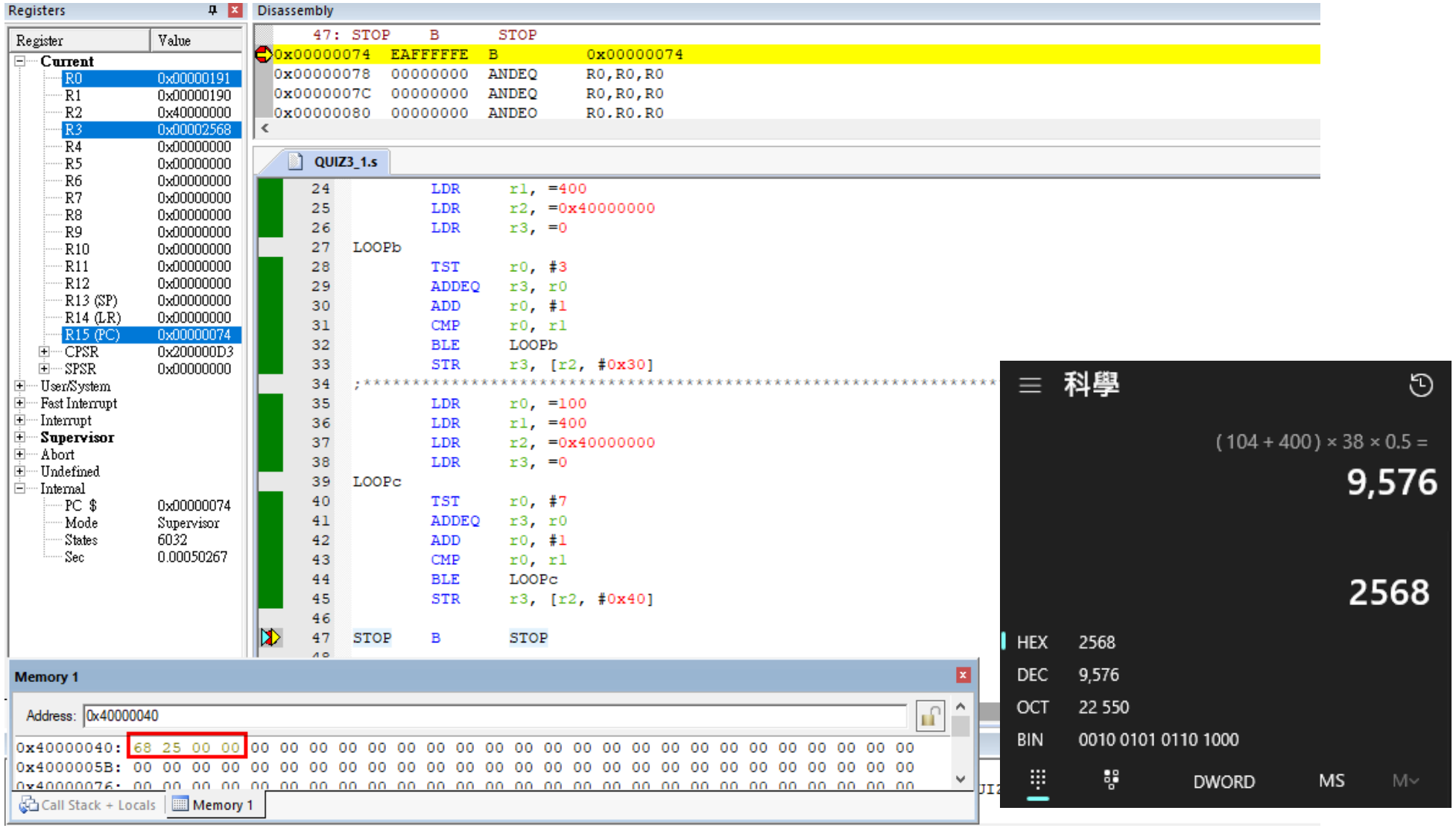
第3次隨堂考-1.(a)


Registers
Register
Value
Current
R0 0x00000191
R1 0x00000190
R2 0x40000000
R3 0x000125F2
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) 0x00000024
CPSR 0x200000D3
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x00000024
Mode Supervisor
States 1810
Sec 0.00015083
Disassembly
23: LDR r0, =100
0x00000024 E3A00064 MOV R0,#0x00000064
24: LDR r1, =400
0x00000028 E3A01E19 MOV R1,#0x00000190
25: LDR r2, =0x40000000
QUIZ3_1.s
1 AREA LIN612450097, CODE, READONLY
2 ENTRY
3 ;*****
4 ;1.(90%)Write a program to calculate the sum of
5 ; (a) numbers
6 ; (b) multiples of 4
7 ; (c) multiples of 8
8 ; between 100 and 400 (inclusive) and leave the word of the sum result respectively in memory
9 ; at addresses 0x40000020, 0x40000030 and 0x40000040.
10 ; (Hint: Be sure to use TST and loops.)
11 ;*****
12 LDR r0, =100
13 LDR r1, =400
14 LDR r2, =0x40000000
15 LDR r3, =0
16 LOOPa
17 ADD r3, r0
18 ADD r0, #1
19 CMP r0, r1
20 BLE LOOPa
21 STR r3, [r2, #0x20]
Memory 1
Address: 0x40000020
0x40000020: F2 25 01 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000003B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack + Locals
Memory 1
科學
(100 + 400) × 301 × 0.5 =
75,250
程式設計人員
1 25F2
HEX 1 25F2
DEC 75,250
OCT 222 762
BIN 0001 0010 0101 1111 0010
DWORD
MS

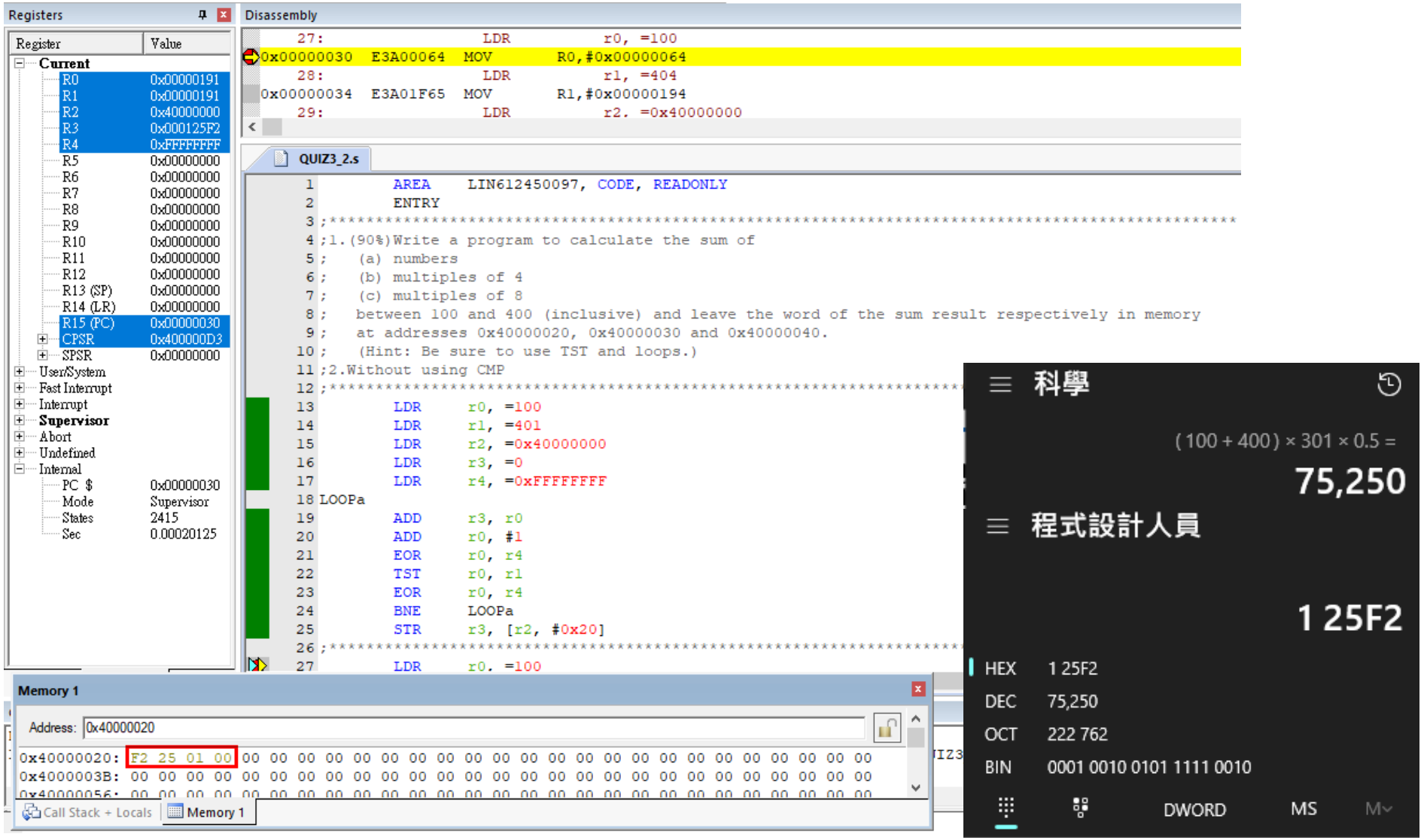
第3次隨堂考-1.(b)


Registers
Register
Value
Current
R0 0x00000191
R1 0x00000190
R2 0x40000000
R3 0x00004A38
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) 0x0000004C
CPSR 0x200000D3
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x0000004C
Mode Supervisor
States 3921
Sec 0.00032675
Disassembly
35: LDR r0, =100
0x0000004C E3A00064 MOV R0,#0x00000064
36: LDR r1, =400
0x00000050 E3A01E19 MOV R1,#0x00000190
37: LDR r2, =0x40000000
QUIZ3_1.s
22 ;*****
23 LDR r0, =100
24 LDR r1, =400
25 LDR r2, =0x40000000
26 LDR r3, =0
27 LOOPb
28 TST r0, #3
29 ADDEQ r3, r0
30 ADD r0, #1
31 CMP r0, r1
32 BLE LOOPb
33 STR r3, [r2, #0x30]
34 ;*****
35 LDR r0, =100
36 LDR r1, =400
37 LDR r2, =0x40000000
38 LDR r3, =0
39 LOOPc
40 TST r0, #7
41 ADDEQ r3, r0
42 ADD r0, #1
Memory 1
Address: 0x40000030
0x40000030: 38 4A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000004B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack + Locals
Memory 1
科學
(100 + 400) × 76 × 0.5 =
19,000
程式設計人員
4A38
HEX 4A38
DEC 19,000
OCT 45 070
BIN 0100 1010 0011 1000
DWORD
MS

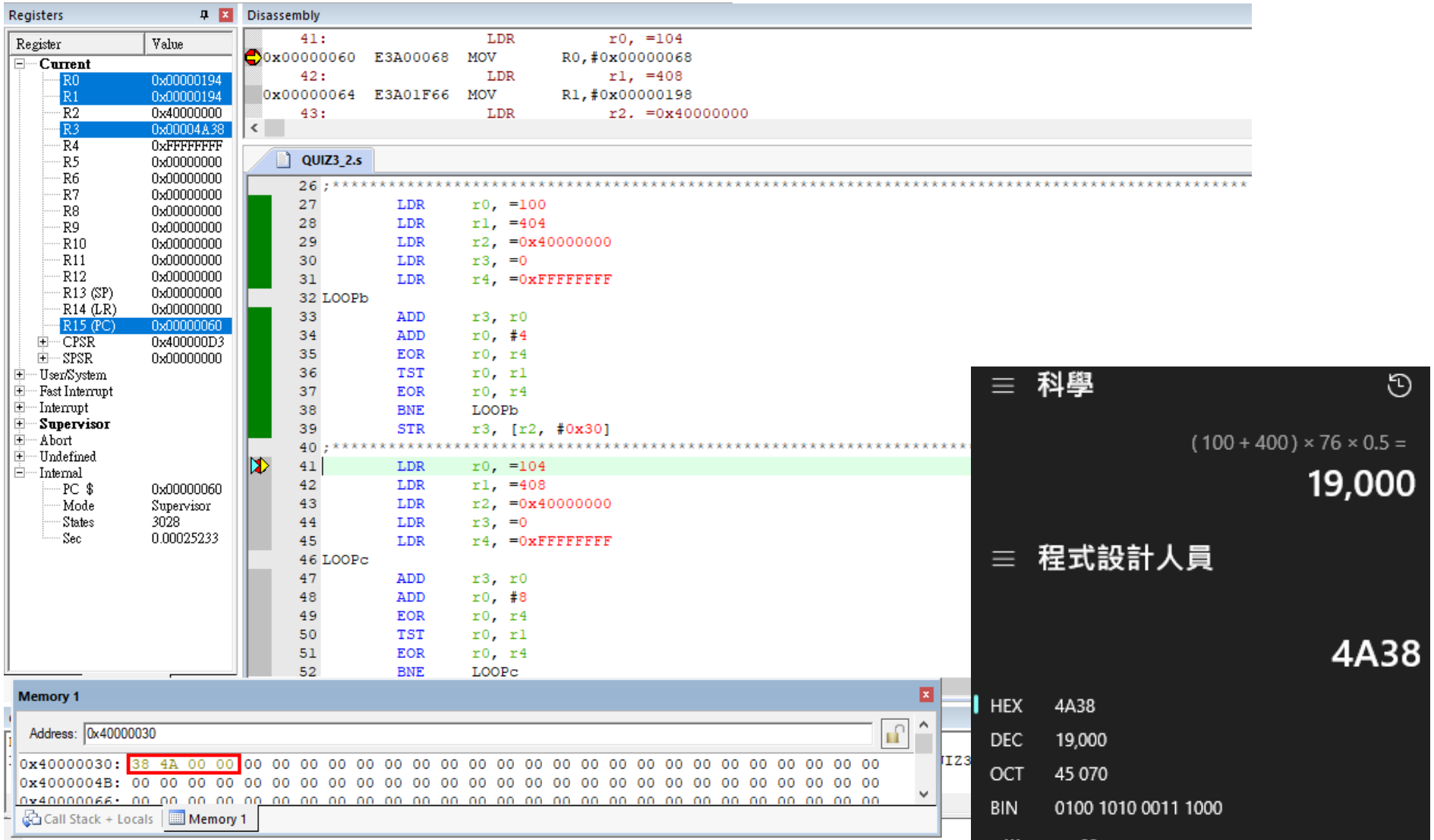
第3次隨堂考-1.(c)


Registers
Register
Value
Current
R0 0x00000191
R1 0x00000190
R2 0x40000000
R3 0x00002568
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) 0x00000074
CPSR 0x200000D3
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x00000074
Mode Supervisor
States 6032
Sec 0.00050267
Disassembly
47: STOP B STOP
0x00000074 EAFFFFFE B 0x00000074
0x00000078 00000000 ANDEQ R0,R0,R0
0x0000007C 00000000 ANDEQ R0,R0,R0
0x00000080 00000000 ANDEQ R0,R0,R0
QUIZ3_1.s
24 LDR r1, =400
25 LDR r2, =0x40000000
26 LDR r3, =0
27 LOOPb
28 TST r0, #3
29 ADDEQ r3, r0
30 ADD r0, #1
31 CMP r0, r1
32 BLE LOOPb
33 STR r3, [r2, #0x30]
34 ;*****
35 LDR r0, =100
36 LDR r1, =400
37 LDR r2, =0x40000000
38 LDR r3, =0
39 LOOPc
40 TST r0, #7
41 ADDEQ r3, r0
42 ADD r0, #1
43 CMP r0, r1
44 BLE LOOPc
45 STR r3, [r2, #0x40]
46
47 STOP B STOP
Memory 1
Address: 0x40000040
0x40000040: 68 25 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000005B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack + Locals
Memory 1
科學
(104 + 400) × 38 × 0.5 =
9,576
2568
HEX 2568
DEC 9,576
OCT 22 550
BIN 0010 0101 0110 1000
DWORD
MS

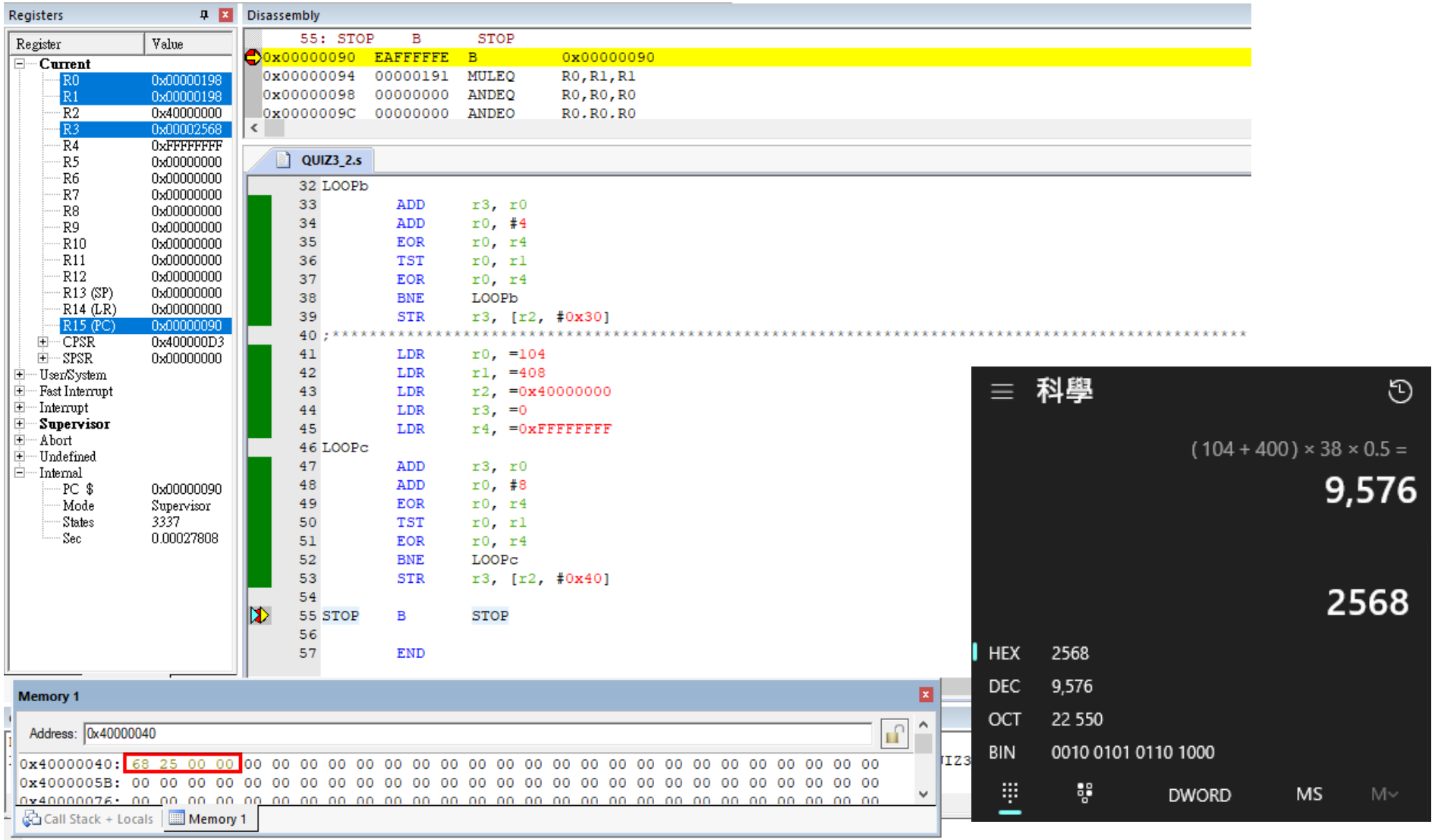
第3次隨堂考-2.(a)


Registers
Register
Value
Current
R0 0x00000191
R1 0x00000191
R2 0x40000000
R3 0x000125F2
R4 0xFFFFFFFF
R5 0x00000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x00000000
R15 (PC) 0x00000030
CPSR 0x400000D3
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x00000030
Mode Supervisor
States 2415
Sec 0.00020125
Disassembly
27: LDR r0, =100
0x00000030 E3A00064 MOV R0,#0x00000064
28: LDR r1, =404
0x00000034 E3A01F65 MOV R1,#0x00000194
29: LDR r2, =0x40000000
QUIZ3_2.s
1 AREA LIN612450097, CODE, READONLY
2 ENTRY
3 ;*****
4 ;1.(90%)Write a program to calculate the sum of
5 ; (a) numbers
6 ; (b) multiples of 4
7 ; (c) multiples of 8
8 ; between 100 and 400 (inclusive) and leave the word of the sum result respectively in memory
9 ; at addresses 0x40000020, 0x40000030 and 0x40000040.
10 ; (Hint: Be sure to use TST and loops.)
11 ;2.Without using CMP
12 ;*****
13 LDR r0, =100
14 LDR r1, =401
15 LDR r2, =0x40000000
16 LDR r3, =0
17 LDR r4, =0xFFFFFFFF
18 LOOPa
19 ADD r3, r0
20 ADD r0, #1
21 EOR r0, r4
22 TST r0, r1
23 EOR r0, r4
24 BNE LOOPa
25 STR r3, [r2, #0x20]
26 ;*****
27 LDR r0, =100
Memory 1
Address: 0x40000020
0x40000020: F2 25 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000003B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack + Locals
Memory 1
科學
(100 + 400) × 301 × 0.5 =
75,250
程式設計人員
1 25F2
HEX 1 25F2
DEC 75,250
OCT 222 762
BIN 0001 0010 0101 1111 0010
DWORD
MS

第3次隨堂考-2.(b)


Registers
Register
Value
Current
R0 0x00000194
R1 0x00000194
R2 0x40000000
R3 0x00004A38
R4 0xFFFFFFFF
R5 0x00000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x00000000
R15 (PC) 0x00000060
CPSR 0x400000D3
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x00000060
Mode Supervisor
States 3028
Sec 0.00025233
Disassembly
41: LDR r0, =104
0x00000060 E3A00068 MOV R0,#0x00000068
42: LDR r1, =408
0x00000064 E3A01F66 MOV R1,#0x00000198
43: LDR r2, =0x40000000
QUIZ3_2.s
26 ;*****
27 LDR r0, =100
28 LDR r1, =404
29 LDR r2, =0x40000000
30 LDR r3, =0
31 LDR r4, =0xFFFFFFFF
32 LOOPb
33 ADD r3, r0
34 ADD r0, #4
35 EOR r0, r4
36 TST r0, r1
37 EOR r0, r4
38 BNE LOOPb
39 STR r3, [r2, #0x30]
40 ;*****
41 LDR r0, =104
42 LDR r1, =408
43 LDR r2, =0x40000000
44 LDR r3, =0
45 LDR r4, =0xFFFFFFFF
46 LOOPc
47 ADD r3, r0
48 ADD r0, #8
49 EOR r0, r4
50 TST r0, r1
51 EOR r0, r4
52 BNE LOOPc
Memory 1
Address: 0x40000030
0x40000030: 38 4A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000004B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack + Locals
Memory 1
科學
(100 + 400) × 76 × 0.5 =
19,000
程式設計人員
4A38
HEX 4A38
DEC 19,000
OCT 45 070
BIN 0100 1010 0011 1000
DWORD
MS

第3次隨堂考-2.(c)


Registers
Register
Value
Current
R0 0x00000198
R1 0x00000198
R2 0x40000000
R3 0x00002568
R4 0xFFFFFFFF
R5 0x00000000
R6 0x00000000
R7 0x00000000
R8 0x00000000
R9 0x00000000
R10 0x00000000
R11 0x00000000
R12 0x00000000
R13 (SP) 0x00000000
R14 (LR) 0x00000000
R15 (PC) 0x00000090
CPSR 0x400000D3
SPSR 0x00000000
User/System
Fast Interrupt
Interrupt
Supervisor
Abort
Undefined
Internal
PC $ 0x00000090
Mode Supervisor
States 3337
Sec 0.00027808
Disassembly
55: STOP B STOP
0x00000090 EAFFFFFE B 0x00000090
0x00000094 00000191 MULEQ R0,R1,R1
0x00000098 00000000 ANDEQ R0,R0,R0
0x0000009C 00000000 ANDEQ R0,R0,R0
QUIZ3_2.s
32 LOOPb
33 ADD r3, r0
34 ADD r0, #4
35 EOR r0, r4
36 TST r0, r1
37 EOR r0, r4
38 BNE LOOPb
39 STR r3, [r2, #0x30]
40 ;*****
41 LDR r0, =104
42 LDR r1, =408
43 LDR r2, =0x40000000
44 LDR r3, =0
45 LDR r4, =0xFFFFFFFF
46 LOOPc
47 ADD r3, r0
48 ADD r0, #8
49 EOR r0, r4
50 TST r0, r1
51 EOR r0, r4
52 BNE LOOPc
53 STR r3, [r2, #0x40]
54
55 STOP B STOP
56
57 END
Memory 1
Address: 0x40000040
0x40000040: 68 25 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x4000005B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
Call Stack + Locals
Memory 1
科學
(104 + 400) × 38 × 0.5 =
9,576
2568
HEX 2568
DEC 9,576
OCT 22 550
BIN 0010 0101 0110 1000
DWORD
MS

1128正課複習

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.)

Hamming Code補充題型(1/4)

1. Get received checksum bits c_0, c_1, c_2, c_3 from the received 12 bits and respectively store in R0, R1, R2, R3.

(1) Get bit 0 from the received 12 bits (R4) and store it in R0.

(Store 12 bits into R0→R0 = 0 or 1→ c_0 = 0 or 1)

(2) Get bit 1 from the received 12 bits (R4) and store it in R1.

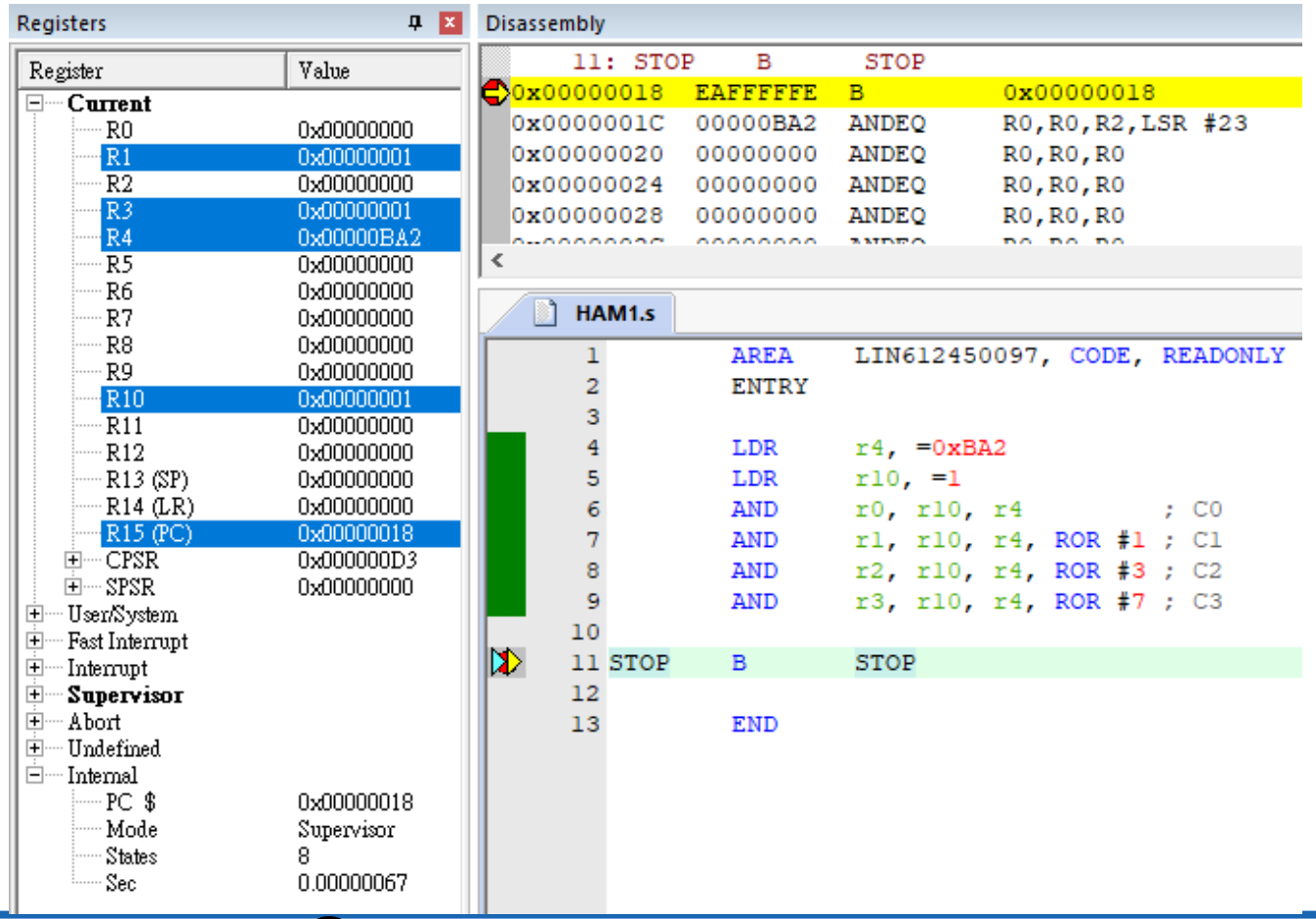
(3) Get bit 3 from the received 12 bits (R4) and store it in R2.

(4) Get bit 7 from the received 12 bits (R4) and store it in R3.

Hamming Code補充題型(1/4)

- 假設Original 8-bit value = 0xB5；

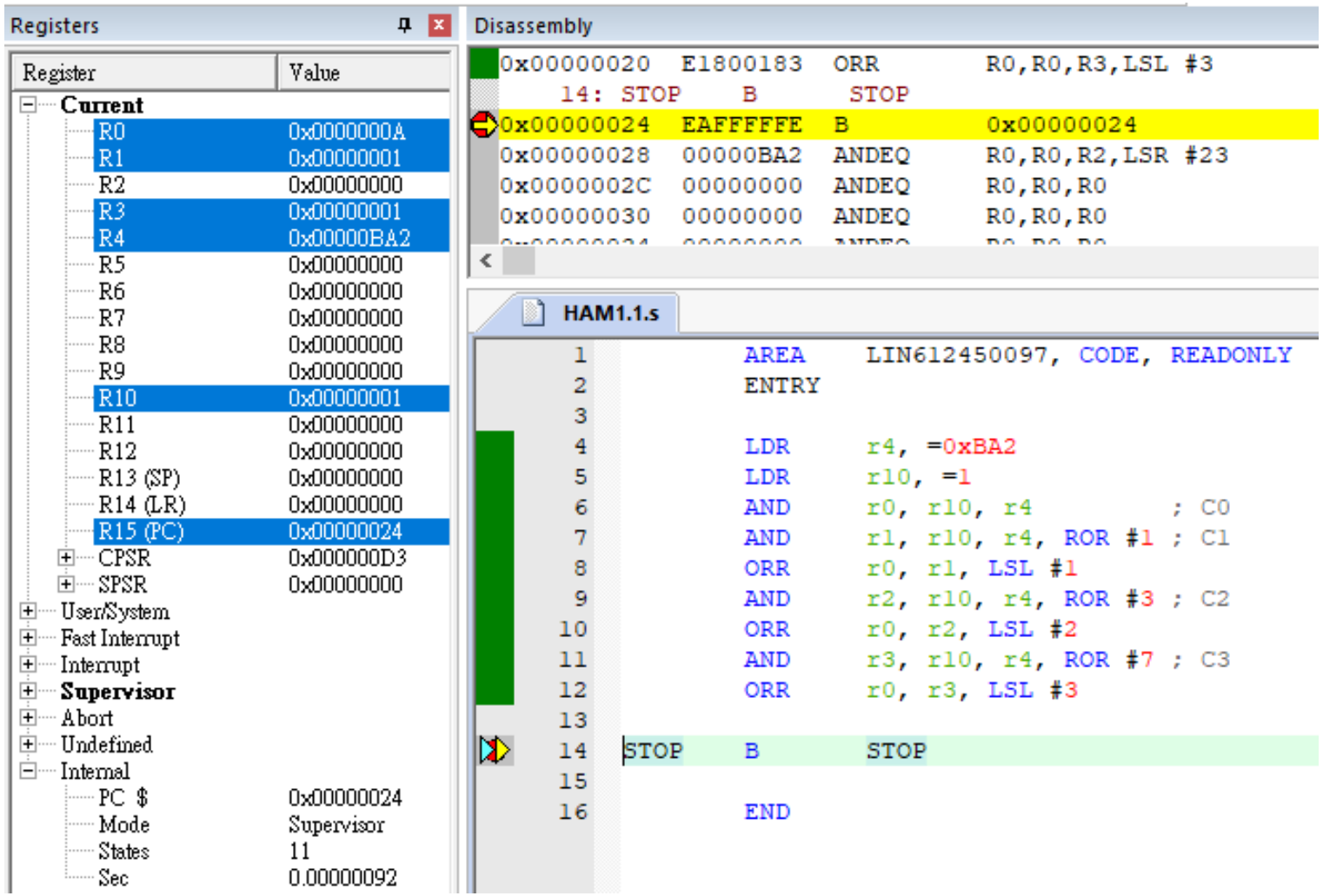
  Modified 8-bit value = 0xBA2(d0的值為錯的：1→0)



Hamming Code補充題型(1/4)

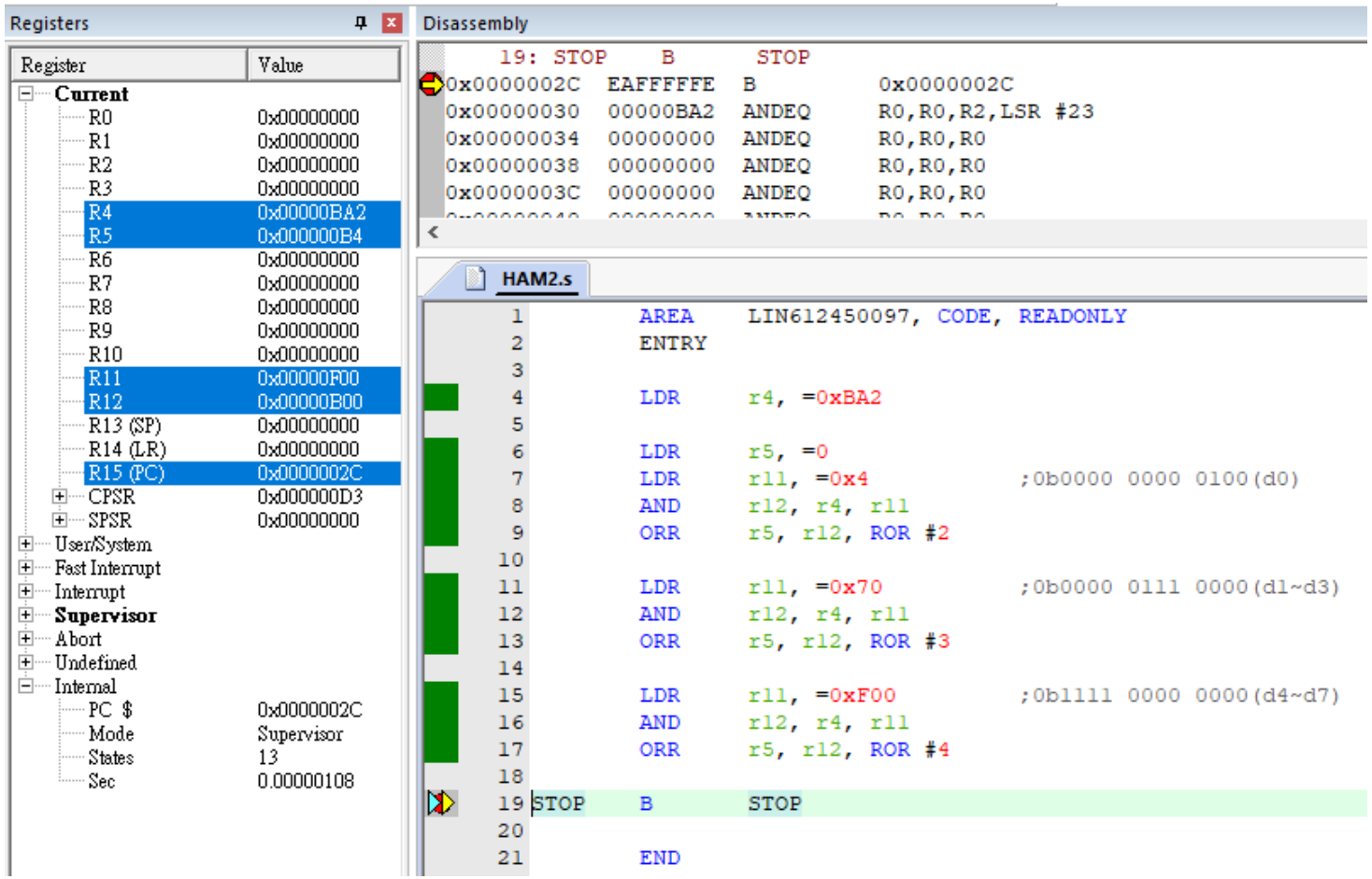
1. Get received checksum bits c_0, c_1, c_2, c_3 from the received 12 bits.
   (1) Get bit 0 from the received 12 bits (R4) and store it in bit 0 of R0.
   (Store 12 bits into R0→R0 = 0 or 1→ c_0 = 0 or 1)
   (2) Get bit 1 from the received 12 bits (R4) and store it in bit 1 of R0.
   (3) Get bit 3 from the received 12 bits (R4) and store it in bit 2 of R0.
   (4) Get bit 7 from the received 12 bits (R4) and store it in bit 3 of R0.

Hamming Code補充題型(1/4)



Hamming Code補充題型(2/4)

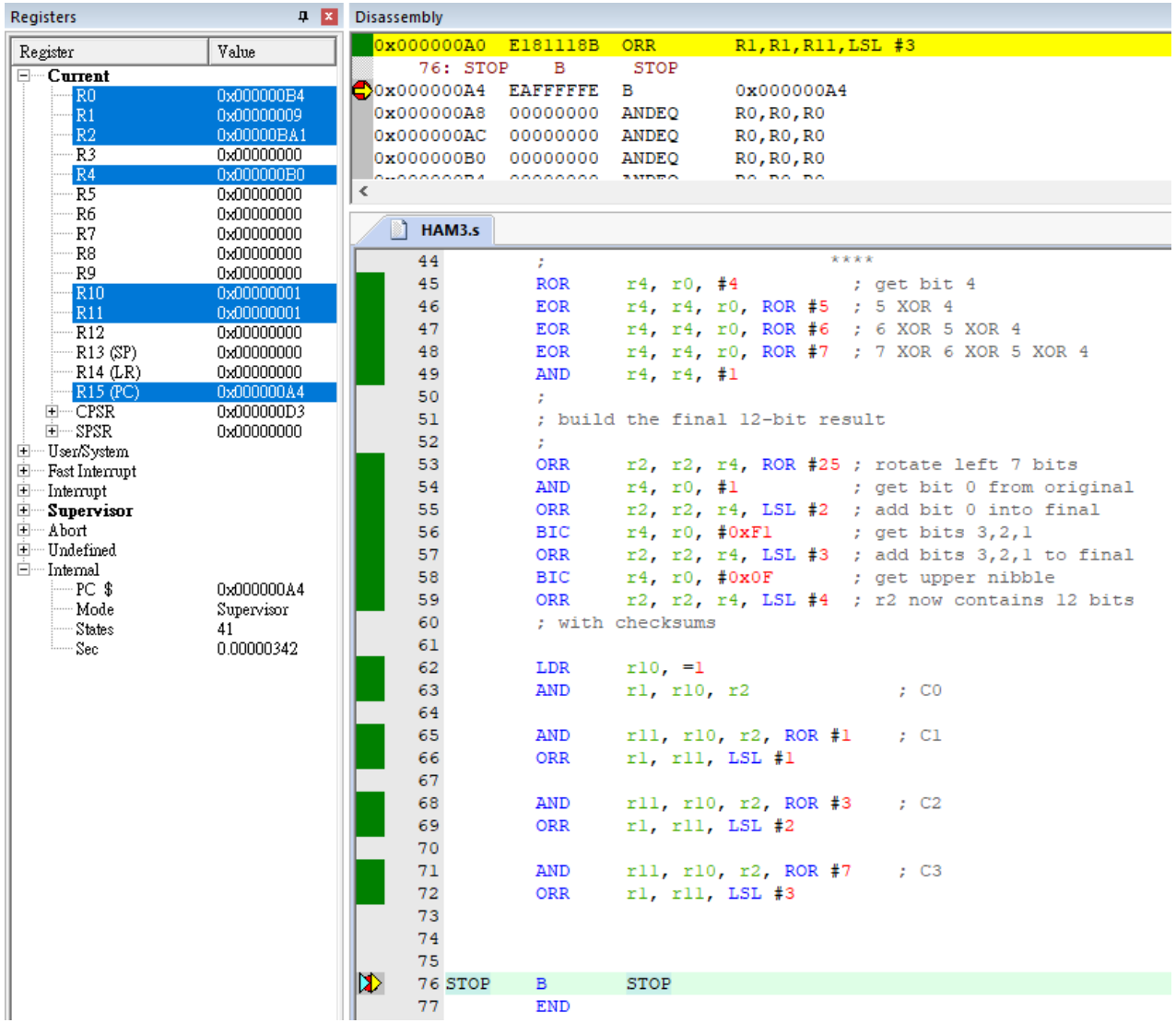
2. Get the received data bits from the received 12 bits(get d7~d0 from the received 12 bits(R4)) and store into r5.



Hamming Code補充題型(3/4)

3. Compute checksum bits c0, c1, c2, c3 (computed checksum bits c0, c1, c2, c3) from the received data bits and respectively store in bit 0~3 of r1.

Hamming Code補充題型(3/4)



Hamming Code補充題型(4/4)

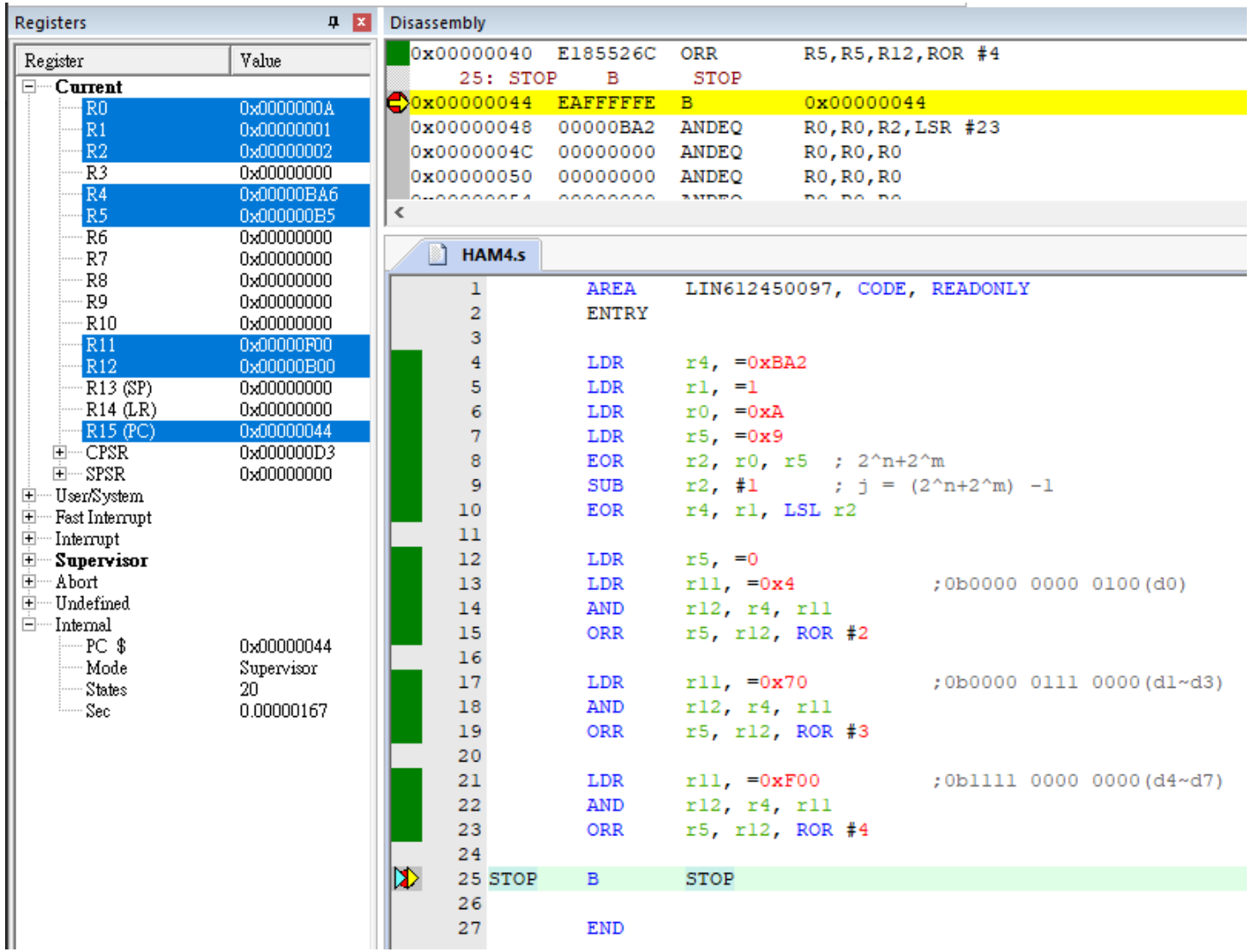
4. Compare received with computed c0, c1, c2, c3.

Compare R0, R5 to see the number of different bits.

(R0 = 0xB；R5 = 0x9 → Two bits different → j = 2)

1) No bit different(correct data bits received)
2) One bits different(checksum bit itself incorrect and data bits correct)
3) Two bits different(one of received data bits is incorrect d_j) → $j = (2^n + 2^m) - 1$
4) Three bits different(暫時不討論)
5) Four bits different(暫時不討論)

Hamming Code補充題型(4/4)



Q&A

Thanks for your attention !!