

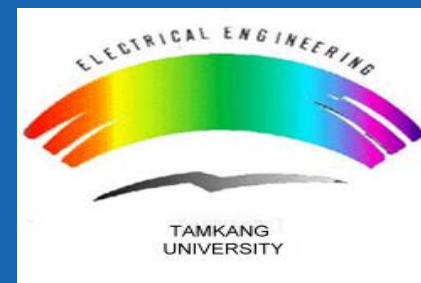
第10次組語實習課

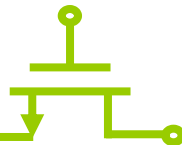
學生：林培瑋

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.)





❖ 1121正課複習	3
❖ 第3次隨堂考	10
❖ 1128正課複習	18

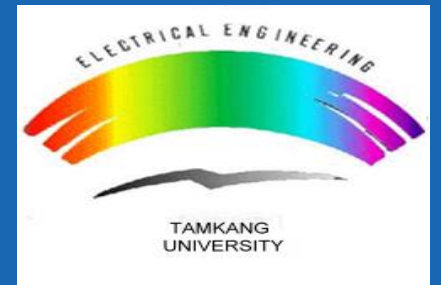


1121正課複習

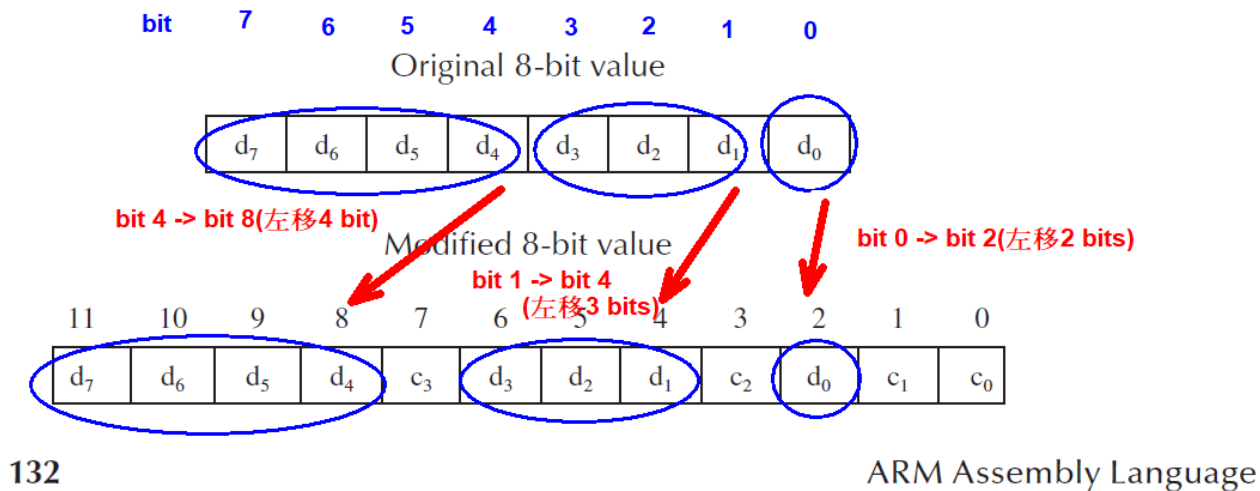
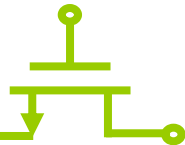
2023 Advanced Mixed-Operation System (AMOS) Lab.



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



```

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
LDR r5, =0xFFFFFFFF
BIC r4, r0, r5
;
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
    
```

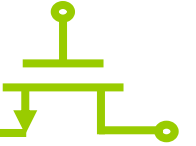
可以寫成

建議寫成

AND r4, r0, #0xE

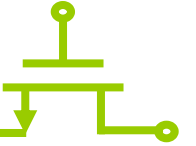
AND r4, r0, #0xF0

因為已知前3個Bytes為0



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 \rightarrow R0 = 0 or 1 $\rightarrow c_0 = 0$ 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



Registers

Register	Value
Current	
R0	0x000000B5
R1	0x000000A0
R2	0x000000BA6
R3	0x00000000
R4	0x000000B0
R5	0x00000001
R6	0x00000000
R7	0x00000001
R8	0x00000000
R9	0x00000001
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (\$F)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x0000009C
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x0000009C
Mode	Supervisor
States	41
Sec	0.00000342

Disassembly

```

69: STOP B STOP
0x0000009C EAffFFFE B 0x0000009C
0x000000A0 AA55AAB5 BGE 0x0156AB7C
0x000000A4 00000000 ANDEQ R0,R0,R0
0x000000A8 00000000 ANDEQ R0,R0,R0

```

HAM.s

```

52 ; build the final 12-bit result
53 ;
54 ORR r2, r2, r4, ROR #25 ; rotate left 7 bits
55 AND r4, r0, #1 ; get bit 0 from original
56 ORR r2, r2, r4, LSL #2 ; add bit 0 into final
57 BIC r4, r0, #0xF1 ; get bits 3,2,1
58 ORR r2, r2, r4, LSL #3 ; add bits 3,2,1 to final
59 BIC r4, r0, #0x0F ; get upper nibble
60 ORR r2, r2, r4, LSL #4 ; r2 now contains 12 bits
61 ; with checksums
62
63 MOV r5, #1
64 AND r6, r5, r2 ; C0
65 AND r7, r5, r2, ROR #1 ; C1
66 AND r8, r5, r2, ROR #3 ; C2
67 AND r9, r5, r2, ROR #7 ; C3
68
69 STOP B STOP
70
71 arraya
72 DCB 0xB5
73 DCB 0xAA
74 DCB 0x55
75 DCB 0xAA
76
77 END

```

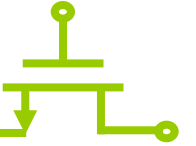
Memory 1

Address: 0x000000A0

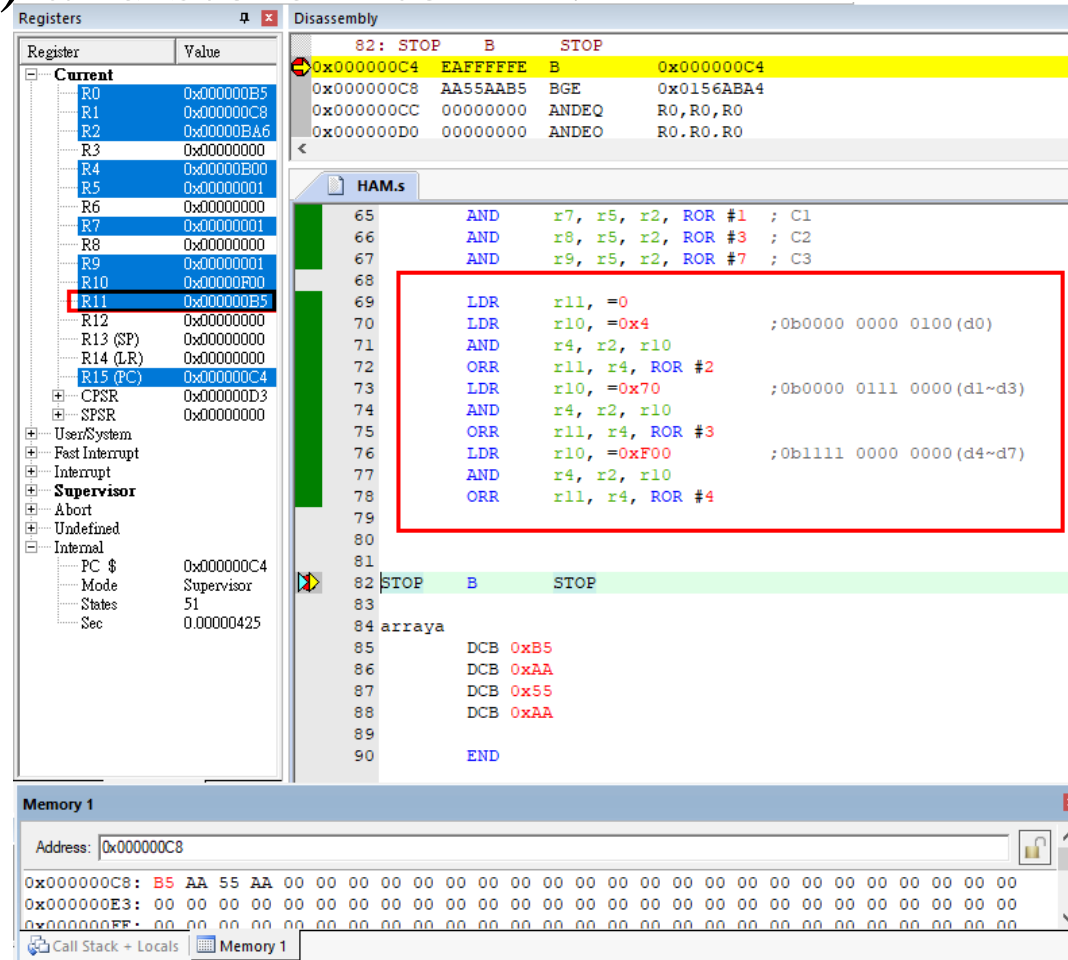
0x000000A0: B5 AA 55 AA 00

0x000000BB: 00

0x000000D6: 00



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



Registers

Register	Value
R0	0x000000B5
R1	0x000000C8
R2	0x000000A6
R3	0x00000000
R4	0x00000000
R5	0x00000001
R6	0x00000000
R7	0x00000001
R8	0x00000000
R9	0x00000001
R10	0x00000000
R11	0x000000B5
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x000000C4
CPSR	0x000000D3
SPSR	0x00000000
UserSystem	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x000000C4
Mode	Supervisor
States	51
Sec	0.00000425

Disassembly

```

82: STOP B STOP
0x000000C4 EAffffff B 0x000000C4
0x000000C8 AA55AAB5 BGE 0x0156ABA4
0x000000CC 00000000 ANDEQ R0,R0,R0
0x000000D0 00000000 ANDEQ R0,R0,R0

HAM.s
65 AND r7, r5, r2, ROR #1 ; C1
66 AND r8, r5, r2, ROR #3 ; C2
67 AND r9, r5, r2, ROR #7 ; C3
68
69 LDR r11, =0
70 LDR r10, =0x4 ; 0b0000 0000 0100 (d0)
71 AND r4, r2, r10
72 ORR r11, r4, ROR #2
73 LDR r10, =0x70 ; 0b0000 0111 0000 (d1~d3)
74 AND r4, r2, r10
75 ORR r11, r4, ROR #3
76 LDR r10, =0xF00 ; 0b1111 0000 0000 (d4~d7)
77 AND r4, r2, r10
78 ORR r11, r4, ROR #4
79
80
81
82 STOP B STOP
83
84 arraya
85 DCB 0xB5
86 DCB 0xAA
87 DCB 0x55
88 DCB 0xAA
89
90 END
  
```

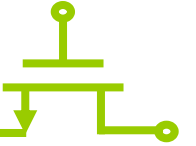
Memory 1

Address: 0x000000C8

```

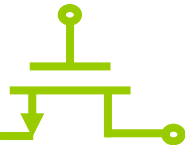
0x000000C8: B5 AA 55 AA 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000E3: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x000000FF: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
  
```





3. Compute checksum bits c_0, c_1, c_2, c_3 (computed checksum bits c_0, c_1, c_2, c_3) from the received data bits.





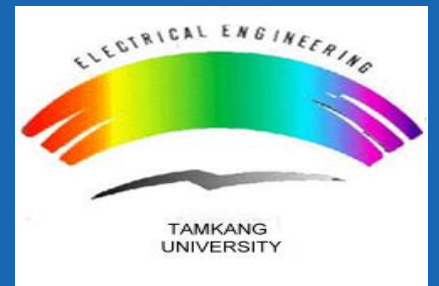
4. Compare **received** with **computed** c_0, c_1, c_2, c_3 .
 - 1) No bit different(correct data bits received)
 - 2) One bits different(checksum bit itself incorrect and data bits correct)
 - 3) Two bits different(one of received data bits is incorrect d_j) $\rightarrow j = (2^n + 2^m) - 1$
 - 4) Three bits different(暫時不討論)
 - 5) Four bits different(暫時不討論)

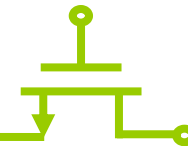
第3次隨堂考

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.)





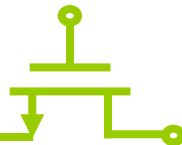
→ 共有3題，一題30分。(若符合第二題規定+10分)

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
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 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x40000030: 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
0x40000040: 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
          
```

科學

$$(100 + 400) \times 301 \times 0.5 =$$

75,250

程式設計人員

1 25F2

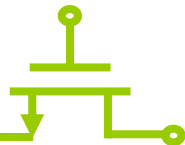
HEX 1 25F2

DEC 75,250

OCT 222 762

BIN 0001 0010 0101 1111 0010

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
43          CMP      r0, r1
          
```

Memory 1

Address: 0x40000030

0x40000030:	38 4A 00 00	00 00
0x4000004B:	00 00	
0x40000066:	00 00	

Call Stack + Locals
Memory 1

科學

$$(100 + 400) \times 76 \times 0.5 =$$

19,000

程式設計人員

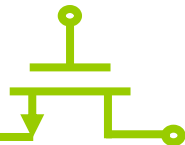
4A38

HEX	4A38
DEC	19,000
OCT	45 070
BIN	0100 1010 0011 1000

DWORD
MS
Mv



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
UserSystem	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x00000074
Mode	Supervisor
States	6032
Sec	0.00050267

Disassembly

Address	Instruction	Comment
47: STOP	B	STOP
0x00000074	EAffFFFFE	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
0x4000005B:	00 00 00 00	00 00
0x40000076:	00 00 00 00	00 00

Call Stack + Locals Memory 1

科學

$$(104 + 400) \times 38 \times 0.5 =$$

9,576

2568

HEX

2568

DEC

9,576

OCT

22 550

BIN

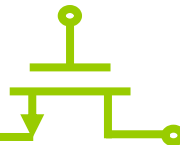
0010 0101 0110 1000

WORD

MS

MS

第3次隨堂考-2.(a)



Register	Value
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
 - State: 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 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 00 00
0x40000056: 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

$$(100 + 400) \times 301 \times 0.5 =$$

75,250

程式設計人員

1 25F2

HEX 1 25F2

DEC 75,250

OCT 222 762

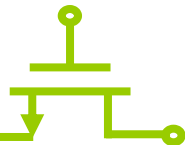
BIN 0001 0010 0101 1111 0010

WORD

MS

Mv

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
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
0x40000066: 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

$$(100 + 400) \times 76 \times 0.5 =$$

19,000

程式設計人員

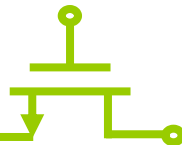
4A38

HEX	4A38
DEC	19,000
OCT	45 070
BIN	0100 1010 0011 1000

DWORD MS Mv



第3次隨堂考-2.(c)



Registers

Register	Value
Current	
R0	0x0000198
R1	0x0000198
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  EAEFFFFE 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
0x4000005B:	00 00 00 00	00 00
0x40000076:	00 00 00 00	00 00

Call Stack + Locals Memory 1

科學

$$(104 + 400) \times 38 \times 0.5 =$$

9,576

2568

HEX

2568

DEC

9,576

OCT

22 550

BIN

0010 0101 0110 1000

DWORD

MS

Mv

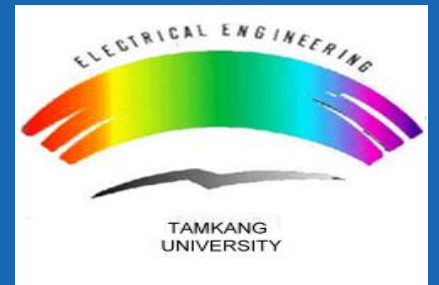


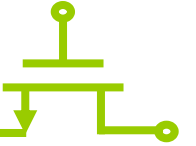
1128正課複習

2023 Advanced Mixed-Operation System (AMOS) Lab.



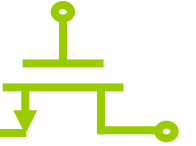
Tamkang University
Department of Electrical and Computer Engineering
No.151, Yingzhuan Rd., Tamsui Dist., New Taipei City 25137, Taiwan (R.O.C.)





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 \rightarrow R0 = 0 or 1 $\rightarrow c_0 = 0$ 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)

Registers

Register	Value
Current	
R0	0x00000000
R1	0x00000001
R2	0x00000000
R3	0x00000001
R4	0x00000BA2
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000001
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000018
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x00000018
Mode	Supervisor
States	8
Sec	0.00000067

Disassembly

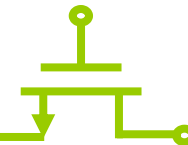
Address	Instruction	Comment
0x00000018	EAFFFFFE B	STOP
0x0000001C	00000BA2 ANDEQ	R0, R0, R2, LSR #23
0x00000020	00000000 ANDEQ	R0, R0, R0
0x00000024	00000000 ANDEQ	R0, R0, R0
0x00000028	00000000 ANDEQ	R0, R0, R0

HAM1.s

```

1      AREA    LIN612450097, CODE, READONLY
2      ENTRY
3
4      LDR     r4, =0xBA2
5      LDR     r10, =1
6      AND     r0, r10, r4          ; C0
7      AND     r1, r10, r4, ROR #1 ; C1
8      AND     r2, r10, r4, ROR #3 ; C2
9      AND     r3, r10, r4, ROR #7 ; C3
10
11     STOP    B      STOP
12
13     END

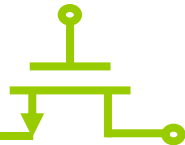
```



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 \rightarrow R0 = 0 or 1 $\rightarrow c_0 = 0$ 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)



Registers

Register	Value
Current	
R0	0x0000000A
R1	0x00000001
R2	0x00000000
R3	0x00000001
R4	0x00000BA2
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000001
R11	0x00000000
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000024
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x00000024
Mode	Supervisor
States	11
Sec	0.00000092

Disassembly

```

0x00000020 E1800183 ORR      R0,R0,R3,LSL #3
14: STOP    B      STOP
0x00000024 EAffffff B      0x00000024
0x00000028 00000BA2 ANDEQ   R0,R0,R2,LSR #23
0x0000002C 00000000 ANDEQ   R0,R0,R0
0x00000030 00000000 ANDEQ   R0,R0,R0
0x00000034 00000000 ANDEQ   R0,R0,R0

```

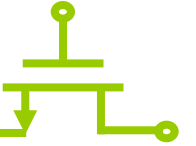
HAM1.1.s

```

1      AREA      LIN612450097, CODE, READONLY
2      ENTRY
3
4      LDR      r4, =0xBA2
5      LDR      r10, =1
6      AND      r0, r10, r4          ; C0
7      AND      r1, r10, r4, ROR #1 ; C1
8      ORR      r0, r1, LSL #1
9      AND      r2, r10, r4, ROR #3 ; C2
10     ORR      r0, r2, LSL #2
11     AND      r3, r10, r4, ROR #7 ; C3
12     ORR      r0, r3, LSL #3
13
14     STOP     B      STOP
15
16     END

```

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.

Registers

Register	Value
Current	
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000BA2
R5	0x00000B4
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000F00
R12	0x00000B00
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x0000002C
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x0000002C
Mode	Supervisor
States	13
Sec	0.00000108

Disassembly

```

19: STOP B STOP
0x0000002C EAffffff B 0x0000002C
0x00000030 00000BA2 ANDEQ R0,R0,R2,LSR #23
0x00000034 00000000 ANDEQ R0,R0,R0
0x00000038 00000000 ANDEQ R0,R0,R0
0x0000003C 00000000 ANDEQ R0,R0,R0
0x00000040 00000000 ANDEQ R0,R0,R0

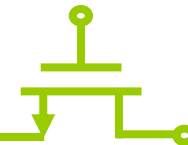
```

HAM2.s

```

1 AREA LIN612450097, CODE, READONLY
2 ENTRY
3
4 LDR r4, =0xBA2
5
6 LDR r5, =0
7 LDR r11, =0x4 ;0b0000 0000 0100 (d0)
8 AND r12, r4, r11
9 ORR r5, r12, ROR #2
10
11 LDR r11, =0x70 ;0b0000 0111 0000 (d1~d3)
12 AND r12, r4, r11
13 ORR r5, r12, ROR #3
14
15 LDR r11, =0xF00 ;0b1111 0000 0000 (d4~d7)
16 AND r12, r4, r11
17 ORR r5, r12, ROR #4
18
19 STOP B STOP
20
21 END

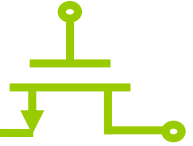
```



3. Compute checksum bits c_0, c_1, c_2, c_3 (computed checksum bits c_0, c_1, c_2, c_3) from the received data bits and respectively store in bit 0~3 of r_1 .



Hamming Code補充題型(3/4)



Registers

Register	Value
Current	
R0	0x000000B4
R1	0x00000009
R2	0x000000A1
R3	0x00000000
R4	0x000000B0
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000001
R11	0x00000001
R12	0x00000000
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x000000A4
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x000000A4
Mode	Supervisor
States	41
Sec	0.00000342

Disassembly

```

0x000000A0 E181118B ORR      R1,R1,R11,LSL #3
76: STOP    B      STOP
0x000000A4 EAFFFFFE B      0x000000A4
0x000000A8 00000000 ANDEQ    R0,R0,R0
0x000000AC 00000000 ANDEQ    R0,R0,R0
0x000000B0 00000000 ANDEQ    R0,R0,R0
0x000000B4 00000000 ANDEQ    R0,R0,R0

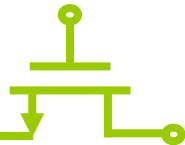
```

HAM3.s

```

44      ; *****
45      ROR    r4, r0, #4      ; get bit 4
46      EOR    r4, r4, r0, ROR #5 ; 5 XOR 4
47      EOR    r4, r4, r0, ROR #6 ; 6 XOR 5 XOR 4
48      EOR    r4, r4, r0, ROR #7 ; 7 XOR 6 XOR 5 XOR 4
49      AND    r4, r4, #1
50      ;
51      ; build the final 12-bit result
52      ;
53      ORR    r2, r2, r4, ROR #25 ; rotate left 7 bits
54      AND    r4, r0, #1      ; get bit 0 from original
55      ORR    r2, r2, r4, LSL #2 ; add bit 0 into final
56      BIC    r4, r0, #0xF1    ; get bits 3,2,1
57      ORR    r2, r2, r4, LSL #3 ; add bits 3,2,1 to final
58      BIC    r4, r0, #0x0F    ; get upper nibble
59      ORR    r2, r2, r4, LSL #4 ; r2 now contains 12 bits
60      ; with checksums
61
62      LDR    r10, =1
63      AND    r1, r10, r2      ; C0
64
65      AND    r11, r10, r2, ROR #1 ; C1
66      ORR    r1, r11, LSL #1
67
68      AND    r11, r10, r2, ROR #3 ; C2
69      ORR    r1, r11, LSL #2
70
71      AND    r11, r10, r2, ROR #7 ; C3
72      ORR    r1, r11, LSL #3
73
74
75
76 STOP    B      STOP
77      END

```



4. Compare **received** with **computed** c_0, c_1, c_2, c_3 .

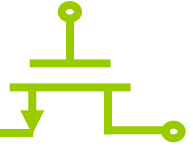
Compare R_0, R_5 to see the number of different bits.

($R_0 = 0xB$; $R_5 = 0x9 \rightarrow$ Two bits different $\rightarrow j = 2$)

- 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) $\rightarrow j = (2^n + 2^m) - 1$
- 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)



Registers

Register	Value
Current	
R0	0x0000000A
R1	0x00000001
R2	0x00000002
R3	0x00000000
R4	0x00000BA6
R5	0x00000E5
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000F00
R12	0x00000B00
R13 (SP)	0x00000000
R14 (LR)	0x00000000
R15 (PC)	0x00000044
CPSR	0x000000D3
SPSR	0x00000000
User/System	
Fast Interrupt	
Interrupt	
Supervisor	
Abort	
Undefined	
Internal	
PC \$	0x00000044
Mode	Supervisor
States	20
Sec	0.00000167

Disassembly

```

0x00000040 E185526C ORR      R5,R5,R12,ROR #4
25: STOP    B      STOP
0x00000044 EAffffff B      0x00000044
0x00000048 00000BA2 ANDEQ    R0,R0,R2,LSR #23
0x0000004C 00000000 ANDEQ    R0,R0,R0
0x00000050 00000000 ANDEQ    R0,R0,R0
0x00000054 00000000 ANDEQ    R0,R0,R0

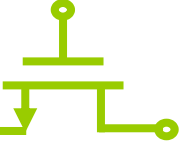
```

HAM4.s

```

1      AREA    LIN612450097, CODE, READONLY
2
3      ENTRY
4
5      LDR      r4, =0xBA2
6      LDR      r1, =1
7      LDR      r0, =0xA
8      LDR      r5, =0x9
9      EOR      r2, r0, r5 ; 2^n+2^m
10     SUB      r2, #1 ; j = (2^n+2^m) -1
11     EOR      r4, r1, LSL r2
12
13     LDR      r5, =0
14     LDR      r11, =0x4 ;0b0000 0000 0100 (d0)
15     AND      r12, r4, r11
16     ORR      r5, r12, ROR #2
17
18     LDR      r11, =0x70 ;0b0000 0111 0000 (d1~d3)
19     AND      r12, r4, r11
20     ORR      r5, r12, ROR #3
21
22     LDR      r11, =0xF00 ;0b1111 0000 0000 (d4~d7)
23     AND      r12, r4, r11
24     ORR      r5, r12, ROR #4
25     STOP     B      STOP
26
27     END

```



Q&A



Thanks for your attention !!