

# 微處理機實習

– 期中考筆記 –

---

**Chun-Hsiang Huang (黃俊翔)**

*Department of Electronic Engineering, Ming Chuan University*

*E-mail: [rissun57@gmail.com](mailto:rissun57@gmail.com)*

---

## Index

I	測驗題	1
1	測驗卷 (一): 微電腦系統基本概念	1
2	測驗卷 (二): LCD 液晶顯示原理	3
3	測驗卷 (三): 加法程式練習	6
4	測驗卷 (四): Atmel 晶片原理與內部架構	13
5	測驗卷 (五): Atmel 暫存器之資料搬移指令	15

---

# 測驗題

PART

I

## 1 測驗卷(一): 微電腦系統基本概念

1. 位元的有號數加法運算，運算完的結果為  $X$ ，則  $X$  在下列何種情況下會產生溢位 (overflow)? ①  $X < -128$  或  $X > +127$  ②  $-128 < X < +127$  ③  $X = -128$  或  $X = +127$  ④  $-128 \leq X \leq +127$ 。

答案：①，當超過邊界時，才會產生溢位。

2. 副程式的呼叫與回返，會使用下列哪一種資料結構？① 堆疊 (stack) ② 貯列 (queue) ③ 樹狀 (tree) ④ 環狀 (ring)。

答案：①，堆疊是先進後出 (LIFO) 的資料結構，適合用於副程式的呼叫與回返。

3. 設一微電腦系統中 ROM 配置在位址  $2000H \sim 4FFFH$ ，而 RAM 配置在位址  $8000H \sim FFFFH$ ，則 ROM 與 RAM 所佔據的空間分別為何？① 4Kbyte、12Kbyte ② 16Kbyte、14Kbyte ③ 12Kbyte、32Kbyte ④ 48Kbyte、64Kbyte。

答案：③，ROM 的大小為  $(4FFFH - 2000H + 1) = 3000H = 12Kbyte$ ，RAM 的大小為  $(FFFFH - 8000H + 1) = 8000H = 32Kbyte$ 。

4. 若有一 5 位元 D/A 轉換器 (D/A converter)，其數位輸入在  $00001B$  時，可以產生的輸出電壓為  $0.2V$ ，則數位輸入為  $11111B$  時的輸出電壓為何？①  $3.2V$  ②  $6.0V$  ③  $6.2V$  ④  $6.4V$ 。

答案：③， $0.2V$  對應的數位輸入為  $00001B = 1$ ，5 位元 D/A 轉換器的最大數位輸入為  $11111B = 31$ ，故  $0.2 \times 31 = 6.2V$ 。

5. 某螢光幕顯示卡的解析度為  $800 \times 600$ ，能提供同時顯示 256 色，則其顯示緩衝記憶體容量至少需多少位元組？① 256K ② 512K ③ 1M ④ 4M。

答案：②， $800 \times 600 = 480000$  像素，每像素 1 byte (256 色)，約為  $480Kbyte$ 。512K 可裝得下。

6. 微電腦之堆疊器都放在哪裡？① I/O ② CPU ③ ROM ④ RAM。

答案：④，堆疊器是用來存放副程式的參數與回傳位址，因此放在 RAM 中。

7. 下列哪種記憶體是不可重覆燒錄？① EEPROM ② EPROM ③ Flash memory ④ PROM。

答案：④，PROM 是不可重覆燒錄的記憶體，而 EPROM 是可以燒錄的記憶體。

8. 位址匯流排有十條，則可定址的範圍為何？①  $0000H \sim 01FFH$  ②  $0000H \sim 02EEH$  ③  $0000H \sim 03FFH$  ④  $0000H \sim 03EEH$ 。

答案：③，位址匯流排有十條，則可定址範圍為  $0 \sim 2^{10} - 1 = 0 \sim 1023 = 03FFH$ 。

9. 位址匯流排包含 24 條線 ( $A0 \sim A23$ )，則可定址空間是多少？① 4MBytes ② 8MBytes ③ 16MBytes ④ 32MBytes。

答案：③， $2^{24} = 16MBytes$ 。

10. 具有 8M Bytes 記憶容量之記憶體，至少需具有多少位址線？① 8 ② 16 ③ 23 ④ 24。

答案：③， $2^{23} = 8M Bytes$ ，因此至少需 23 條位址線。

11. 微電腦每執行一運算指令後，將運算後的各種狀態存於何者？ ① 堆疊 ② 索引暫存器 ③ 累積器 ④ 旗標暫存器。

答案：④，旗標暫存器用來儲存運算後的狀態。

12. 標準的 ASCII 碼是幾位元？ ① 6 位元 ② 7 位元 ③ 8 位元 ④ 9 位元。

答案：②，標準 ASCII 碼為 7 位元。

13. 將記憶位址的一部分規劃為輸入/輸出埠的方法稱為：① 記憶體映射式 I/O ② 間接式 I/O ③ 儲存式 I/O ④ 獨立式 I/O。

答案：①，記憶體映射式 I/O 是將記憶位址的一部分用作 I/O 埠。

14. 下列哪些選項為合法的 8 進制數？ ① 417 ② 128 ③ 119 ④ 255。

答案：①④，8 進制數的每位數字應為 0~7，故 417 和 255 合法。

15. 下列哪些單元屬於 CPU 內部的結構？ ① 控制單元 CU ② 算術單元 AU ③ 邏輯單元 LU ④ 記憶體單元 MU。

答案：①②③，CPU 內部包含控制單元、算術單元和邏輯單元，記憶體單元屬外部。

16. 兩個 4 位數 BCD 碼相加，下列哪些旗標會受到影響？ ① 進位旗標 ② 輔助進位旗標 ③ 溢位旗標 ④ 同位旗標。

答案：①②，BCD 碼相加會影響進位旗標和輔助進位旗標。

17. EPROM 27128，其容量為何？ ① 128kbytes ② 64kbytes ③ 32kbytes ④ 16kbytes。

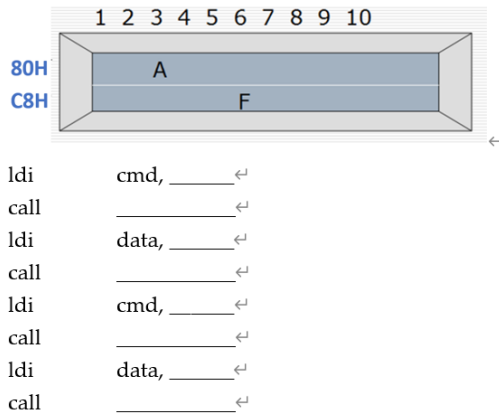
答案：④，27128 表示 128K bits，換算為  $128K/8 = 16Kbytes$ 。

18. 二進位數 0101010101 轉換成十進位數，等於多少？ ① 5 ② 314 ③ 341 ④ 11111。

答案：③， $0101010101_2 = 2^8 + 2^6 + 2^4 + 2^2 + 2^0 = 256 + 64 + 16 + 4 + 1 = 341_{10}$ 。

## 2 測驗卷 (二): LCD 液晶顯示原理

1. 參考圖一 LCD 的字元顯示，請寫出 ATmel 語言程式碼，共計 2 個字元



```
ldi    cmd, _____  
call   _____  
ldi    data, _____  
call   _____  
ldi    cmd, _____  
call   _____  
ldi    data, _____  
call   _____
```

**Descriptio 1.** LCD 字元顯示

答案：

```
ldi    cmd, 0x82    ; 設定顯示位置  
call   wlcd_command ; 寫入顯示位置命令到 LCD  
ldi    data, 0x41   ; 寫入字元 A  
call   wlcd_data    ; 寫入字元到 LCD  
ldi    cmd, 0xc5    ; 設定顯示位置  
call   wlcd_command  
ldi    data, 0x46   ; 寫入字元 F  
call   wlcd_data
```

2. 參考圖二 LCD 的字串顯示，請寫出 atmel 語言程式碼，共計 2 行字串

```
str1_table:
.db      "Ming Chuan Univ.",0x80
str2_table:
.db      "EE37201 Pu-Sheng",0x80
    ldi      cmd, _____
    call     _____
    ldi      ZH, _____
    ldi      ZL, _____
    call     _____
    ldi      cmd, _____
    call     wlcd_command
    ldi      ZH, _____
    ldi      ZL, _____
    call     _____
```

Descriptio 2. LCD 字元顯示

答案：

```
    ldi      cmd, 0x80                ; 設定顯示位置
    call     wlcd_command            ; 寫入顯示位置命令到 LCD
    ldi      ZH, high(2*str1_table) ; 設定 str1 字串位置
    ldi      ZL, low(2*str1_table)
    call     disp_lcd_string ; 顯示字串到 LCD
    ldi      cmd, 0xc0
    call     wlcd_command
    ldi      ZH, high(2*str2_table) ; 設定 str2 字串位置
    ldi      ZL, low(2*str2_table)
    call     disp_lcd_string

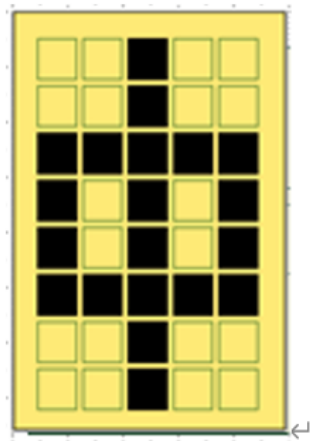
str1_table:
.db      "Ming Chuan University", 0x80

str2_table:
.db      "EE37201 Pu-Sheng", 0x80
```

註解：

； disp\_lcd\_string 是一個函式，會將 Z 指向的字串顯示到 LCD 上。  
； wlcd\_command 是一個函式，會將命令寫入到 LCD 上。  
； str1\_table 和 str2\_table 是兩個字串，分別存放在記憶體中。  
； 具體流程就是，先設定顯示位置，然後將字串寫入到 LCD 上。  
； 其中 str1\_table 和 str2\_table 是兩個字串，  
    他們也有屬於自己的記憶體位置，  
    而利用 ldi 指令將其位置寫入到 Z 寄存器中，  
    需要\*2 是因為 str1\_table 和 str2\_table 是 16 位元的資料，  
    所以需要乘上 2 才能得到正確的位址。

3. 參考圖中 LCDM 字型，請寫出該字型碼，共計八碼



Descriptio 3. LCD 字元顯示

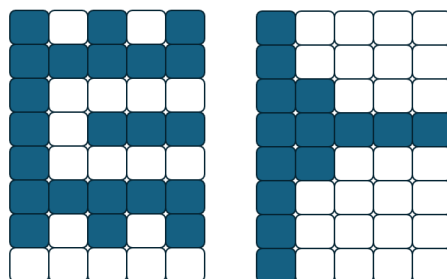
答案：

0x04, 0x04, 0x1F, 0x15, 0x15, 0x1F, 0x04, 0x04; 中

```
char_table:
.db    0x15,0x1F,0x10,0x17,0x10,0x1F,0x15,0x00
      0x10,0x10,0x18,0x1F,0x18,0x10,0x10,0x00
```

Descriptio 4. LCD 字元顯示

4. 以上為 LCDM 兩個字型碼，試繪出這兩個字形碼之造型。



Descriptio 5. LCD 字元顯示

5. 請問 G 的 ASCII 碼為 0x47, g 的 ASCII 碼為 0x67, 8 的 ASCII 碼為 0x38, 空白的 ASCII 碼為 0x20。

### 3 測驗卷 (三): 加法程式練習

1. 以下程式段功能是將兩個 10 進制資料相加，並將其結果以 10 進制格式儲存至 r13、r12、r11 三個暫存器中，請填寫以下空格。

```
bcd2hex_16    0x00,0x00,0x32  → r15= _____ r14= _____ ↵
movw          r6,r14          → r7= _____ r6= _____ ↵
bcd2hex_16    0x00,0x00,0x16  → r15= _____ r14= _____ ↵
movw          r12,r6          → r13= _____ r12= _____
call          add_16bit↵
movw          r14,r10↵
call          hex_bcd16       → r13= _____ r12= _____ r11= _____ ↵
```

#### Descriptio 6. 加法程式練習

答案：

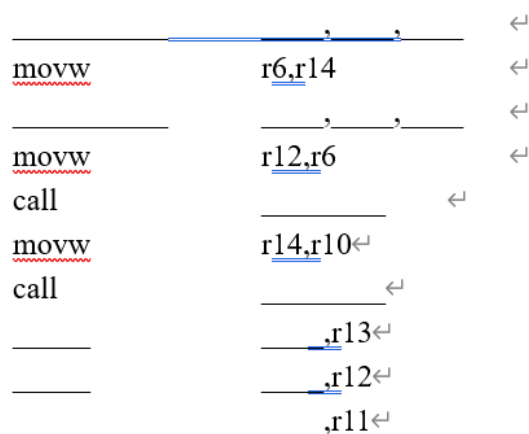
```
bcd2hex_16    0x00, 0x00, 0x32 -> r15 = 0x00, r14 = 0x20
movw          r6, r14          -> r7 = 0x00, r6 = 0x20
bcd2hex_16    0x00, 0x00, 0x16 -> r15 = 0x00, r14 = 0x10
movw          r12, r6          -> r13 = 0x00, r12 = 0x20
call          add_16bit        ; r11 = 0x00, r10 = 0x30
movw          r14, r10        ; r15 = 0x00, r14 = 0x30
call          hex_bcd16        -> r13 = 0x00, r12 = 0x00 , r11 = 0x48
```

註解：

;bcd 的轉換 可以使用工程計算機處理。  
;bcd2hex\_16 會把 在 r13, r12, r11 組成的 bcd 數字轉成十六進制並  
存在 r15, r14。  
;movw 會把 r14, r13 複製到 r6, r7,  
順帶一個容易忽略的觀念是，偶數對偶數才能做 movw。  
;add\_16bit 會把 r15, r14 與 r13, r12 相加，  
並把結果存到 r11, r10。  
;hex\_bcd16 會把 r15, r14 轉成 bcd，  
並把結果存到 r13, r12, r11。



2. 利用巨集指令來計算  $35678 + 21349$ ，並且將結果存放至 mem\_H, mem\_M, mem\_L (十進制加法)



Descriptio 7. 巨集指令加法

答案：

```

-> bcd2hex_16      0x03, 0x56, 0x78 ; r15 = 0x8B, r14 = 0x5E
    movw           r6, r14          ; r7 = 0x8B, r6 = 0x5E
-> bcd2hex_16      0x02, 0x13, 0x49 ; r15 = 0x53, r14 = 0x65
    movw           r12, r6          ; r13 = 0x8B, r12 = 0x5E
-> call            add_16bit         -> r11 = 0xde, r10 = 0xc3
    movw           r14, r10          ; 使其可以被 hex_bcd16 使用
-> call            hex_bcd16         -> r13 = 0x05, r12 = 0x70, r11= 0x27
-> sts             mem_H, r13        ; mem_H = 0x05
-> sts             mem_M, r12        ; mem_M = 0x70
-> sts             mem_L, r11        ; mem_L = 0x27

```

註解：

； bcd2hex\_16 會把 35678 寫入，r13, r12, r11，並且調用 bcd\_hex16  
把 r13, r12, r11 轉成十六進制，放入 r15, r14。  
； hex\_bcd16 會把 r15, r14 轉成 bcd，  
並把結果存到 r13, r12, r11。  
；透過 sts @0, Rd 這個操作，把暫存器寫入記憶體中。

3. 利用模組 `add_16bit` 計算 `793AH + 9B4DH`，並且將結果存放至記憶體 (`ans_H`, `ans_L`)  
(十六進制加法)

```

_____      _____,r15↵
_____      _____,r14↵
_____      _____,r13↵
_____      _____,r12↵
call         add_16bit↵
_____      ans_H,_____↵
_____      ans_L,_____↵
試問執行完本程式之後，ans_H=_____ , ans_L=_____↵

```

#### Descriptio 8. 16 位元加法

答案：

```

w_register    0x79, r15 ; r15 = 0x79
w_register    0x3A, r14 ; r14 = 0x3A
w_register    0x9B, r13 ; r13 = 0x9B
w_register    0x4D, r12 ; r12 = 0x4D

call          add_16bit ; (r15, r14)+(r13, r12)=(r11, r10)
sts           ans_H, r11 ; ans_H= 0x14
sts           ans_L, r10 ; ans_L= 0x87

-> ans_H = 0x14, ans_L = 0x87

```

註解：

- ； `0x793A + 0x9B4D = 0x11487`，但是題目使用 `add_16bit`，也就意味著，最前頭的進位不需要考慮。
- ； 具體實現是，利用 `w_register` 將資料寫入暫存器中，然後調用 `add_16bit`，將結果放入 `r11`, `r10` 中。
- ； 最後使用 `sts @0, Rd` 這個操作，把暫存器寫入記憶體中。
- ； 提供一個解釋，可以視為 `w_register @0, Rd` 的操作，這樣就可以將資料寫入暫存器中。

4. 檢測 count 暫存器與 0x9a 之間關係，倘若 count=0x9a 程式跳至 LOOP\_1 標記，倘若 count>0x9a 程式跳至 LOOP\_2 標記，倘若 count<0x9a 程式跳至 LOOP\_3 標記

Introduce :

要處理這個問題，應該要認識到 AVR ISA 有哪些關於條件判斷的指令，依據其 Flags 不同來決定其判斷的屬性。

這些指令有：

- > Z (Zero Flag): 結果是否為 0
  - BRNE label : Z=0 時跳轉 (Not Equal)
  - BREQ label : Z=1 時跳轉 (Equal)
  
- > N (Negative Flag): 結果最高位是否為 1 (視為有符號數的負數)
  - BRPL label : N=0 時跳轉 (Positive)
  - BRMI label : N=1 時跳轉 (Negative)
  
- > V (Overflow Flag): 有符號數運算是否溢位
  - BRVC label : V=0 時跳轉 (No Overflow)
  - BRVS label : V=1 時跳轉 (Overflow)
  
- > S (Sign Flag,  $S = N \oplus V$ ): 有符號數比較的實際符號
  - BRLT label : S=1 時跳轉 (Less Than, 表示  $A < B$ )
  - BRGE label : S=0 時跳轉 (Greater or Equal, 表示  $A \geq B$ )
  
- > C (Carry Flag): 無符號數運算的進位/借位
  - BRCC/BRSH label : C=0 時跳轉 (No Carry)
  - BRCS/BRL0 label : C=1 時跳轉 (Carry Set)
  
- > H (Half Carry Flag): BCD 運算或低位元組進位
  - BRHS label : H=1 時跳轉 (Half Carry Set)
  - BRL0 label : H=0 時跳轉 (Lower)
  
- > T (Global Flag): 通用位元標記
  - BRTS label : T=1 時跳轉 (T Flag Set)
  - BRTC label : T=0 時跳轉 (T Flag Clear)

比較指令為：

- CP Rd, Rr : 比較兩個暫存器，並更新 Flags
- CPI Rd, K : 比較暫存器與常數，並更新 Flags

答案：

```
    cpi        count, 0x9a ; 比較 count 與 0x9a
    brne       LOOP_a ; count != 0x9a 時跳轉到 loop_a
    jmp        LOOP_1
```

```
LOOP_a:
    brlo       LOOP_b ; count < 0x9a 時跳轉到 loop_2
    jmp        LOOP_2
```

```
LOOP_b:
    jmp        LOOP_3 ; count > 0x9a 時跳轉到 loop_1
```

註解：

；cpi 指令會將 count 與 0x9a 進行比較，並且 cpi 不會特別區分有號數與無號數，所以基本上其跳轉指令使用無號數的比較即可。

5. 利用間接定址 X 索引暫存器將除法小數點 r10，共計 24 位，儲存至記憶體\$0120

```

movw    r8,r6
ldi     count,24
ldi     r27,0x01
ldi     r26,0x20
dvpt_1:
movw    r14,r8
w_register    0,r13
w_register    10,r12
call    mul_16bit
movw    r14,r8
movw    r12,r2
call    div_16bit
        ,r10
        24
brne    dvpt_1
.ENDMACRO

```

#### Descriptio 9. 除法程式練習

答案：

```

ldi     count, 24 ; count = 24
ldi     r27, 0x01
ldi     r26, 0x20 ; r26 = 0x20, r27 = 0x01 , x=0x0120

```

```

dvpt_1:
....
....
st      X+, r10    ; 將 r10 寫入記憶體中，並且 x++
dec     count      ; count--
brne    dvpt_1     ; count != 0 時跳轉到 dvpt_1

```

註解：

；要找 第 24 位的除法小數點，所以需要使用 count 來做迴圈，  
count 需要從 24 開始，然後每次迴圈減少 1。  
；將小數點 r10 寫入記憶體中 24 位，至 X = \\$0120。

；再次複習一下，st, sts, lds, lpm, ld 差異：  
；st @0, Rd：將暫存器 Rd 的內容寫入到記憶體中。

透過 X, Y, Z 暫存器來做間接定址。

- ； sts @0, Rd : 將暫存器 Rd 的內容寫入到記憶體中，  
直接針對記憶體位址。
- ； lds Rd, @0 : 將記憶體中的內容讀取到暫存器中，  
直接針對記憶體位址。
- ； lpm Rd, Z+ : 將記憶體中的內容讀取到暫存器中，  
透過 Z 暫存器來做間接定址。
- ； ld Rd, Y+ : 將記憶體中的內容讀取到暫存器中，  
透過 X, Y, 暫存器來做間接定址。

## 4 測驗卷 (四): Atmel 晶片原理與內部架構

1. Atmel 晶片採用 4MHz 石英振盪器，振盪週期為何？機械週期為何？以下程式執行時間為何？所占記憶體大小為何？

振盪週期: 0.25  $\mu$  s

機械週期: 1  $\mu$  s

程式執行時間: 41  $\mu$  s

所占記憶體大小: 36 bytes

2. 參考以下程式，程式執行完畢後，以下暫存器及記憶體內容為何？

r0 = 23h  
 r1 = 46h  
 r2 = 57h  
 r3 = 2ah  
 r4 = 86h ←

r26 = 3dh    r27 = 02h  
 r28 = 37h    r29 = 02h ←

\$0232 = 2ah  
 \$0233 = 86h  
 \$0234 = 6ah  
 \$0235 = 2ah ←

\$023a = 6ah  
 \$023b = 2ah  
 \$023c = 86h ←

ldi	r27, \$02			
ldi	r26, \$3a			
ldi	r29, \$02			
ldi	r28, \$30	f6h	023dH	
ld	r0, Y+	e5h	023cH	
ld	r1, Y+	d7h	023bH	
ldi	r28, \$33	c8h	023aH	
ld	r2, Y	b3h	0239H	
ld	r3, -Y	adh	0238H	
ldi	r28, \$33	9ch	0237H	
lds	r4, \$0236	86h	0236H	
st	Y+, r4	7bh	0235H	
sts	\$0235, r3	6ah	0234H	
ldi	count, 3	57h	0233H	
loop:		2ah	0232H	
ld	buf_1, Y+	46h	0231H	
st	X+, buf_1	23h	0230H	←
dec	count			
brne	loop			

3. Atmel 晶片採用 1MHz 石英振盪器，振盪週期為何？機械週期為何？圖中程式執行時間為何？所占記憶體大小為何？

delay_3:	<u>ldi</u>	buf_3, 250
delay_2:	<u>ldi</u>	buf_4, 250
delay_1:	<u>dec</u>	buf_4
	<u>brne</u>	delay_1
	<u>dec</u>	buf_3
	<u>brne</u>	delay_2

Descriptio 10. 指令執行時間

振盪週期: 1  $\mu$  s, 機械週期: 1  $\mu$  s, 執行時間: 188250us;  $((1+2)250+2)*250$ , 記憶體大小: 12 bytes

4. ATmega168 中 flash ROM 佔有多大記憶體空間？RAM 佔有多大記憶體空間？EEPROM 佔有多大記憶體空間？

Flash ROM: 16kbytes, RAM: 16kbytes, EEPROM: 512bytes

5. 說明何謂 RISC? RISC 晶片之特性為何？

RISC (精簡指令集計算機) 指令很少：131, 每一個指令佔一個記憶體的單位 (2 byte), 每一個指令執行時間只佔一個機械週期 (machine cycle)

6. 記憶體映射問題

- (a) ATmega168 通用暫存器佔有多大記憶體空間？位址由 0x0000 ~ 0x001F
- (b) ATmega168 IO 暫存器佔有多大記憶體空間？位址由 0x0020 ~ 0x005F
- (c) ATmega168 擴充 IO 暫存器佔有多大記憶體空間？位址由 0x0060 ~ 0x00FF
- (d) ATmega168 記憶體佔多大記憶體空間？位址由 0x0100 ~ 0x04FF

7. Embedded MicroController Unit (EMCU)、Embedded MicroProcessor Unit (EMPU)、Digital Signal Processor (EDSP)、Embedded System on Chip (ESOC), 說明各晶片之間的差異為何？

- EMCU: 整合 CPU、記憶體、周邊於單晶片，適合控制應用
- EMPU: 高性能處理器核心，需外接記憶體和周邊，(具有作業系統 OS)
- EDSP: 嵌入式數位信號處理器 (具有浮點數運算能力)
- ESOC: 嵌入式晶片上系統 (可重新規劃晶片內部結構)



## 5 測驗卷 (五): Atmel 暫存器之資料搬移指令

在 AVR 架構下, R\_0 ~ R\_15 不能被立即數直接存取, 而 R\_16 ~ R\_31 可以被立即數直接存取。

- (X) ldi r16, #97
- (X) ldi r28, 9d
- (O) ldi r18, \$9d
- (O) ldi r17, 0x9d
- (X) ldi r15, 0b01001000
- (X) ldi r24, 257
- (X) ldi r3, 25
- (X) cpi r9, \$9d
- (O) cp r2, r4
- (O) add r17, r7
- (X) andi r7, \$7a
- (O) ori r17, \$7a
- (O) movw r6, r26
- (X) movw r7, r26
- (X) movw r7, r27
- (O) lpm r7, Z+
- (O) lpm r17, -Z
- (X) lpm r7, Y+
- (X) lpm r17, -Y
- (O) adiw ZH:ZL, 2
- (O) adiw YH:YL, 63
- (X) adiw XH:XL, 250
- (O) adiw r24, 1
- (X) adiw r4, 1
- (X) ld X+, reg
- (X) st -Y, code
- (X) sts count, \$02ff
- (O) lds data, \$01ff
- (X) sts cmd, \$0100
- (O) ld r2, -Y
- (O) lds r4, mem
- (X) dec mem
- (O) inc reg

```

ldi        ZH,high(2*str_table)
ldi        ZL,low(2*str_table)
lpm        r1,Z+
lpm
lpm        r2,+Z
adiw       ZH:ZL,2
lpm        r3,+Z
str_table:
.db        "cd",0xaa,0xbb,0xcc,0xdd,0xee,0xff

```

上述程式執行完畢後，

r0= 0x64, r1= 0x63, r2= 0xaa, r3= 0xdd

注意：*lpm* 後為空就是讀取 *Z* 寄存器的內容至 *r\_0*