



MICROCHIP

# dsPIC30F Peripheral Module

## High-Speed 10-bit ADC

# Session Agenda

- ❖ Module Overview
- ❖ ADPCFG – A/D Port Configuration Register
- ❖ ADCON1 – A/D Control Register 1
- ❖ ADCON2 – A/D Control Register 2
- ❖ ADCON3 – A/D Control Register 3
- ❖ ADCHS – A/D Input Select Register
- ❖ ADCSSL – A/D Input Scan Select Register
- ❖ ADC Lab ( Hand-On Season )

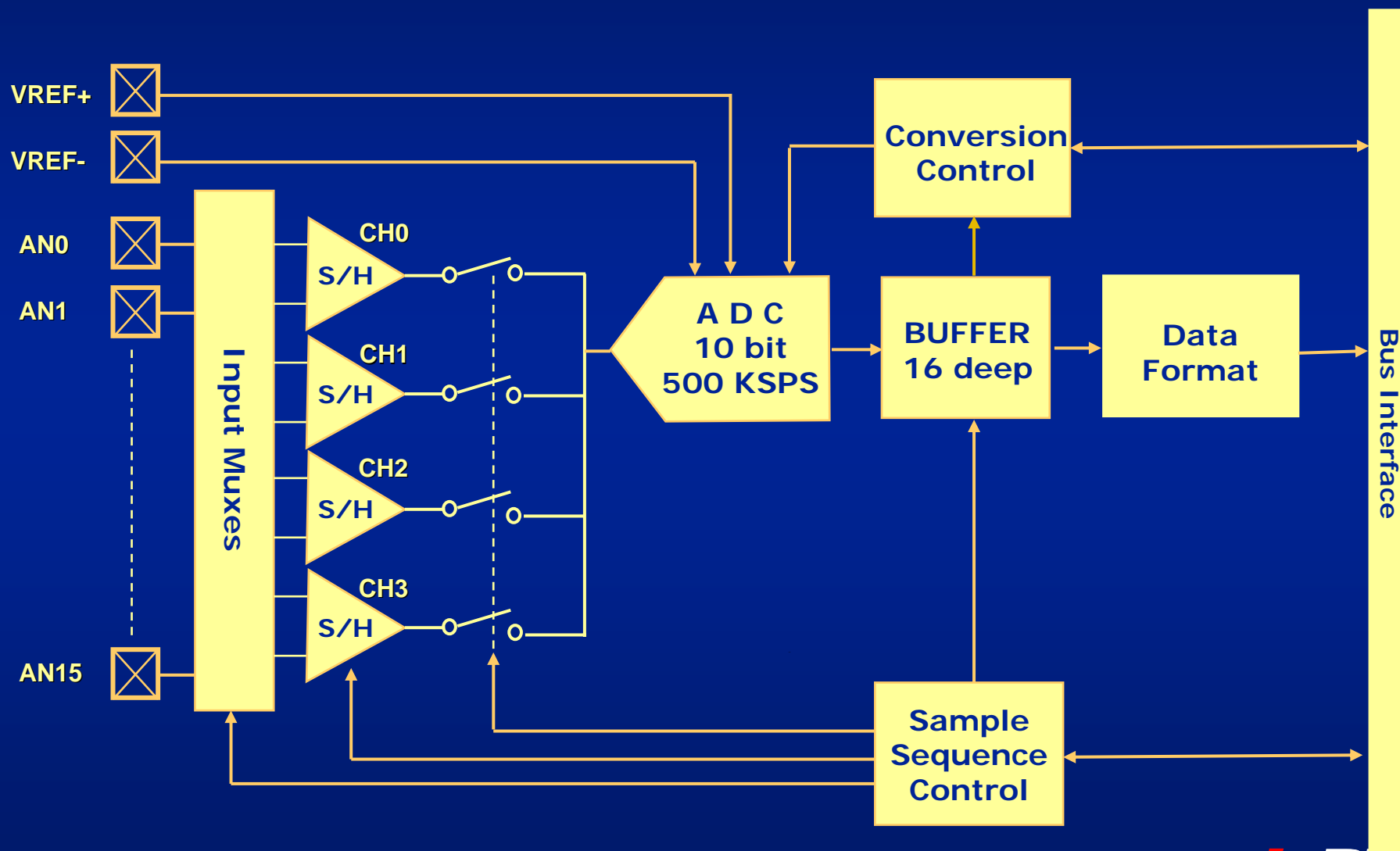
# 10-bit 高速 AD 轉換器

- ❖ dsPIC30F4011 10-bit ADC
  - ◆ Up to 500 Ksps (2 $\mu$ S) conversion speed
  - ◆ External voltage reference pin
  - ◆ Four unipolar difference S/H amplifiers
  - ◆ Simultaneous sampling of up to four input
  - ◆ Automatic Channel Scan Mode
  - ◆ Selectable conversion trigger source
  - ◆ 16 word conversion result buffer
  - ◆ Four result alignment options
  - ◆ Operation during CPU IDLE and SLEEP mode



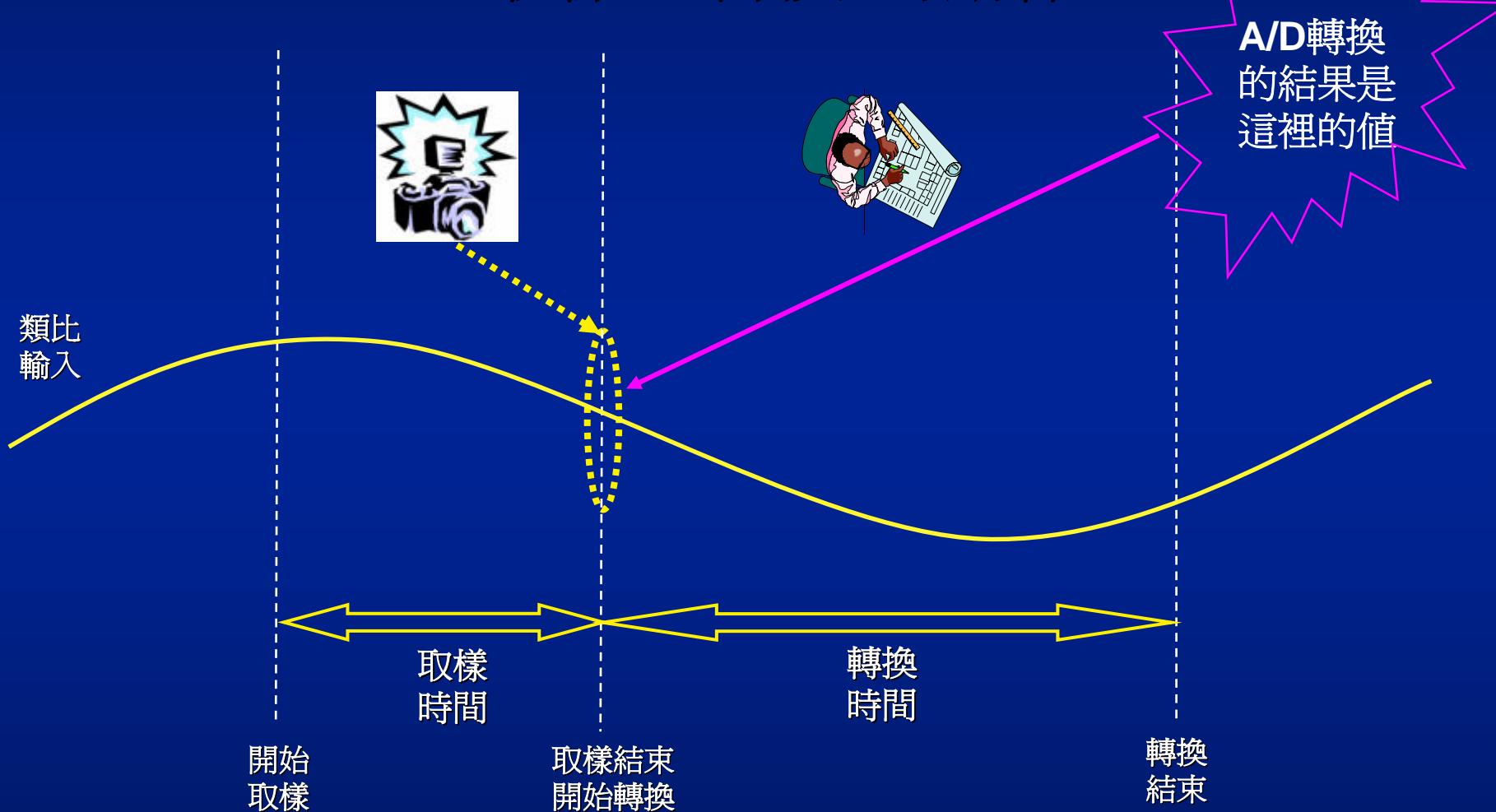
MICROCHIP

# 10-bit A/D 方塊圖



# A/D 轉換器

## 取樣 / 轉換 的動作





MICROCHIP

# A/D 轉換器 ADPCFG 暫存器

定義：  
類比輸入腳位設定



MICROCHIP

# ADPCFG 暫存器

每一類比輸入腳位都可以單獨設定



PCFG<15:0> - 選擇類比輸入腳位

- PCFGx = 1 → 數位輸出入功能
- PCFGx = 0 → 類比輸入功能

- 零件編號不同，類比輸入的接腳數也會不一樣
- 重置後(**Reset**)，預設腳位的功能為類比輸入
- 每一類比輸入腳位都可以單獨設定
- 設成類比輸入腳以後，**I/O** 讀取的結果為 **0**



MICROCHIP

# A/D 轉換器

## ADCON1 暫存器

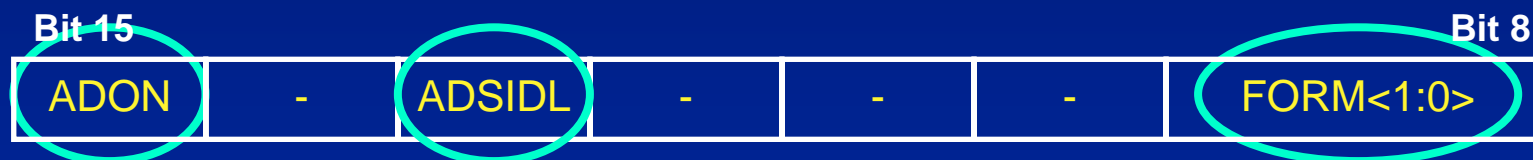
定義：

取樣模式 & 轉換觸發模式 & 資料輸出格式



# ADCON1 暫存器 (MSB)

ADCON1



**ADON** – 設為 1 時，將 ADC 轉換功能致能

**ADSIDL** – 設為 1 時，ADC 在 IDLE 模式下會關閉

**FORM<1:0>** - AD轉換後資料的輸出格式選擇

- 11 = 有號小數 (Signed fractional)
- 10 = 無號小數 (Fractional)
- 01 = 有號整數 (Signed integer)
- 00 = 整數 (Integer)

# ADC 輸出格式

## 16-bit

	b15	b14	←-----→	b1	b0
有號 小數	—		d09, d08, d07, d06, d05, d04, d03, d02, d01, d00,	0, 0, 0, 0, 0, 0	
小數			d09, d08, d07, d06, d05, d04, d03, d02, d01, d00,	0, 0, 0, 0, 0, 0	
有號 整數	—	—	d09, d09, d09, d09, d09, d09, d09, d08, d07, d06, d05, d04, d03, d02, d01, d00		
整數			0, 0, 0, 0, 0, 0, d09, d08, d07, d06, d05, d04, d03, d02, d01, d00		

# ADCON1 暫存器 (LSB)

ADCON1



**SSRC<2:0>** - 啟動AD轉換的觸發信號來源選擇

111 = 使用內部時序設定取樣時間及轉換時間(自動轉換)

(需參考到 **ADCON3** 暫存器的設定)

011 = 馬達控制 PWM 間隔結束時，結束取樣ADC開始轉換

(需參考到 **SEVTCMP** 暫存器的設定) 馬達專用的dsPIC才有此功能

010 = Timer 3 計時比較完成後，結束取樣ADC開始轉換

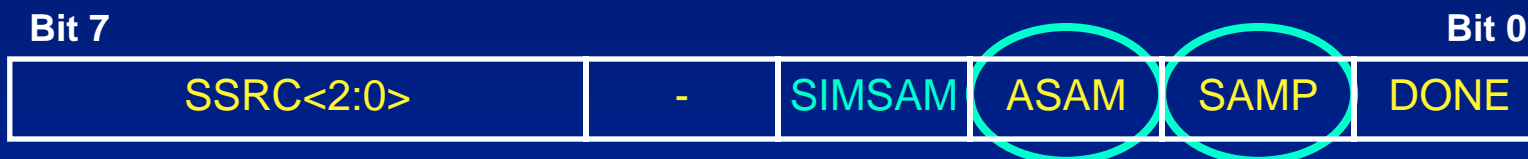
001 = INT0 腳位電位轉態時，結束取樣ADC開始轉換

(需參考到 **INTCON2** <INT0EP> 位元的設定)

000 = 手動轉換，SAMP=1 時取樣，清除 SAMP 位元後轉換

# ADCON1 暫存器 (LSB)

ADCON1



## ASAM :

- 設為 1 時，當上次AD轉完後成後，立即自動取樣 (SAMP位元會自動設為 1)，
- 設為 0 時，採手動取樣模式，將 SAMP位元設定為 1 時取樣。

## SAMP :

- SAMP 寫入 1 時就開始取樣；SAMP 清為 0 後，採樣工作結束，然後 AD 開始轉換 (當DONE=1或ADIF=1 時表示轉換完成)
- 當 SSRC 不為 000 時，所指定到的 Trigger Source 會設定 SAMP = 1，並在取樣結束後自動的將 SAMP 清為 “0”，並進行 AD 轉換

# ADCON1 暫存器 (LSB)

ADCON1

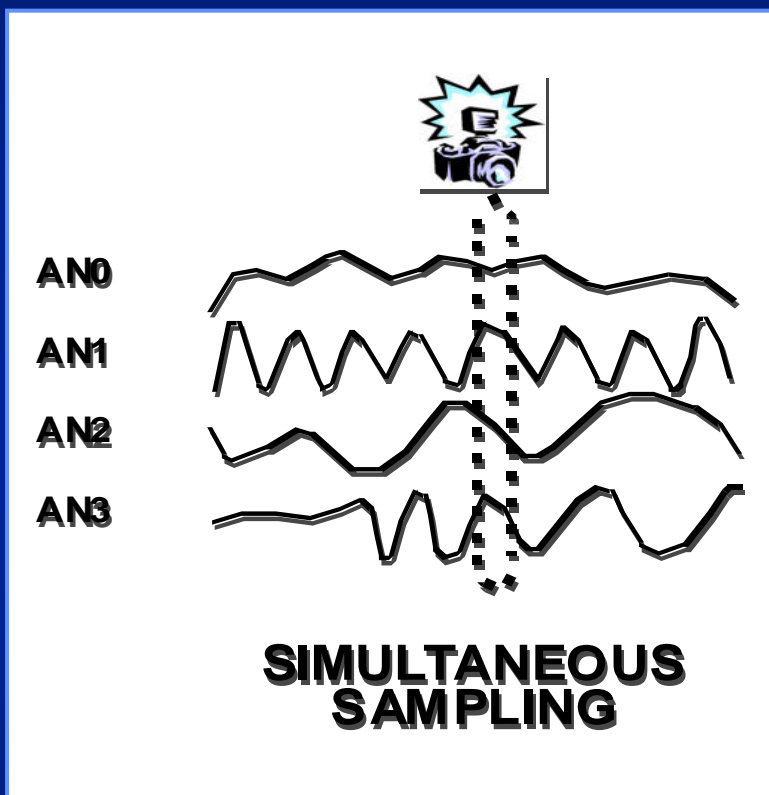


**DONE** : AD 轉換狀態指示位元

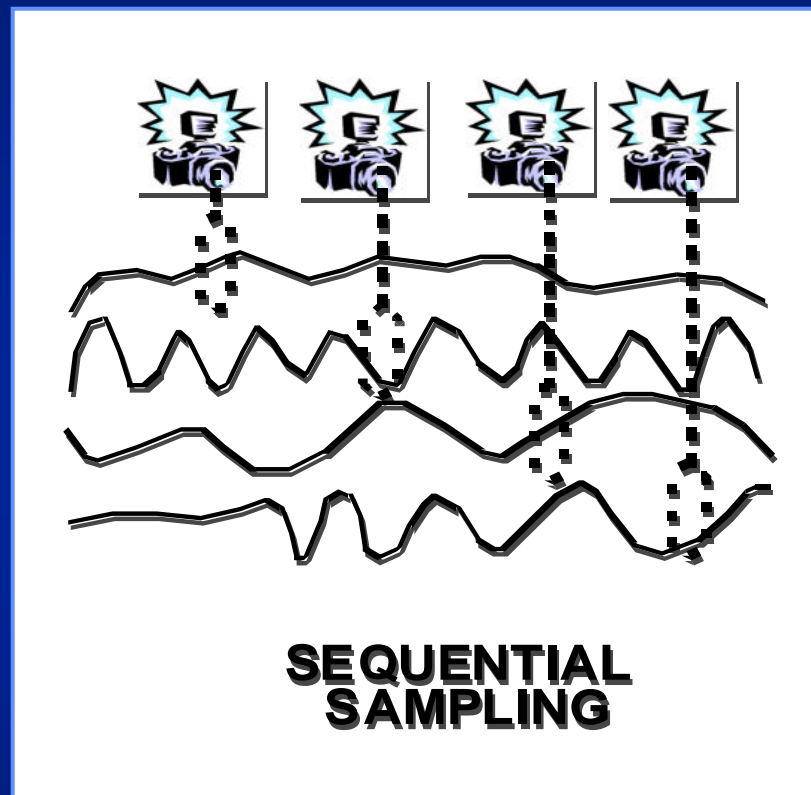
等於 1 時，A/D 轉換完成，該位元可以用軟體清除或新的轉換動作啟動時也會被清為 0

**SIMSAM** : 同時取樣設定位元，SIMASM = “1” 時 CH0 ~ CH3 同時取樣 (CHPS=1x)，SIMASM = “1” 時 CH0 ~ CH1 同時取樣 (CHPS=01)，SIMASM = “0” 時各自順序取樣

# 兩種取樣方式



同時取樣模式

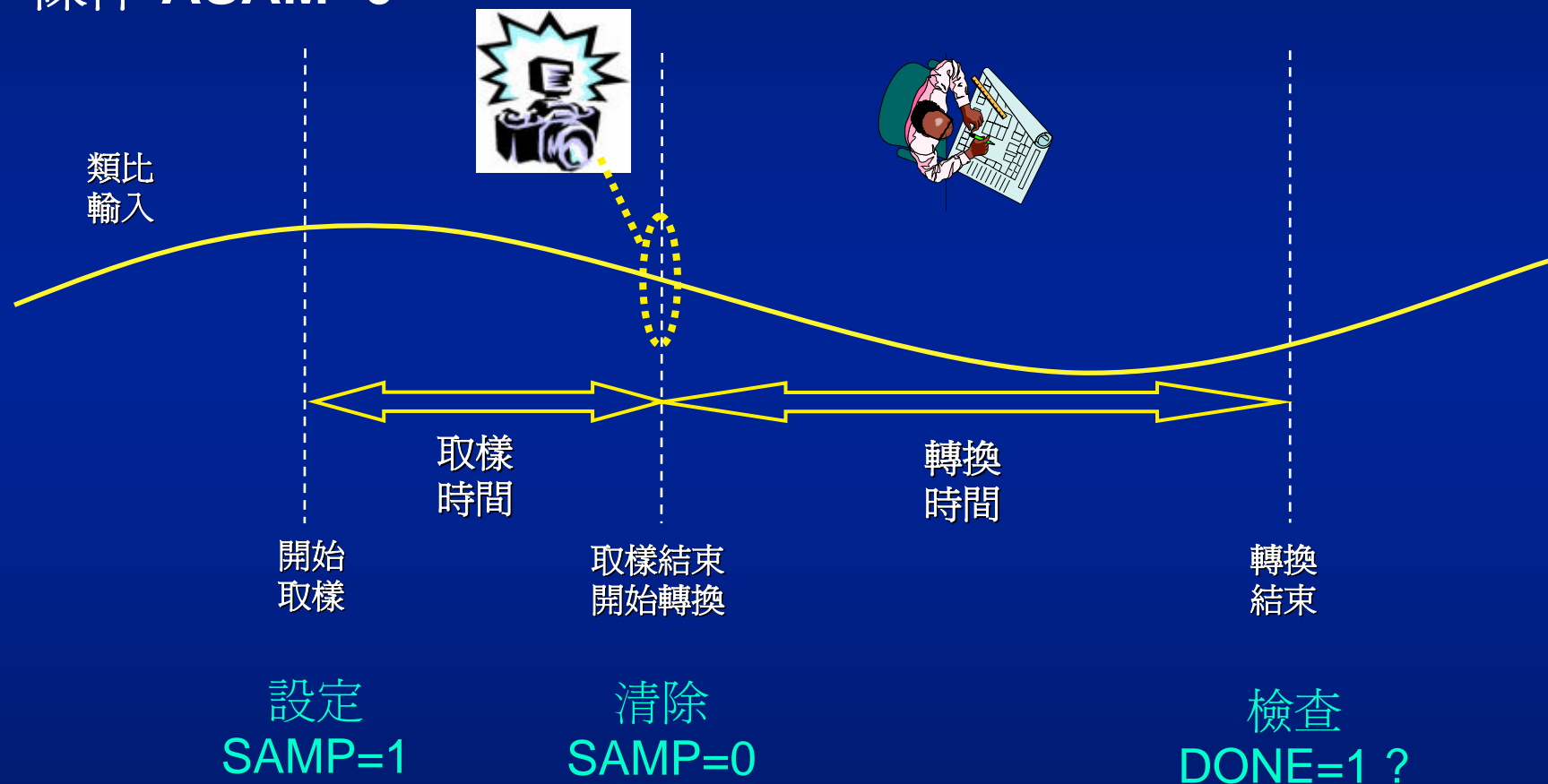


依序取樣模式

# A/D 轉換器

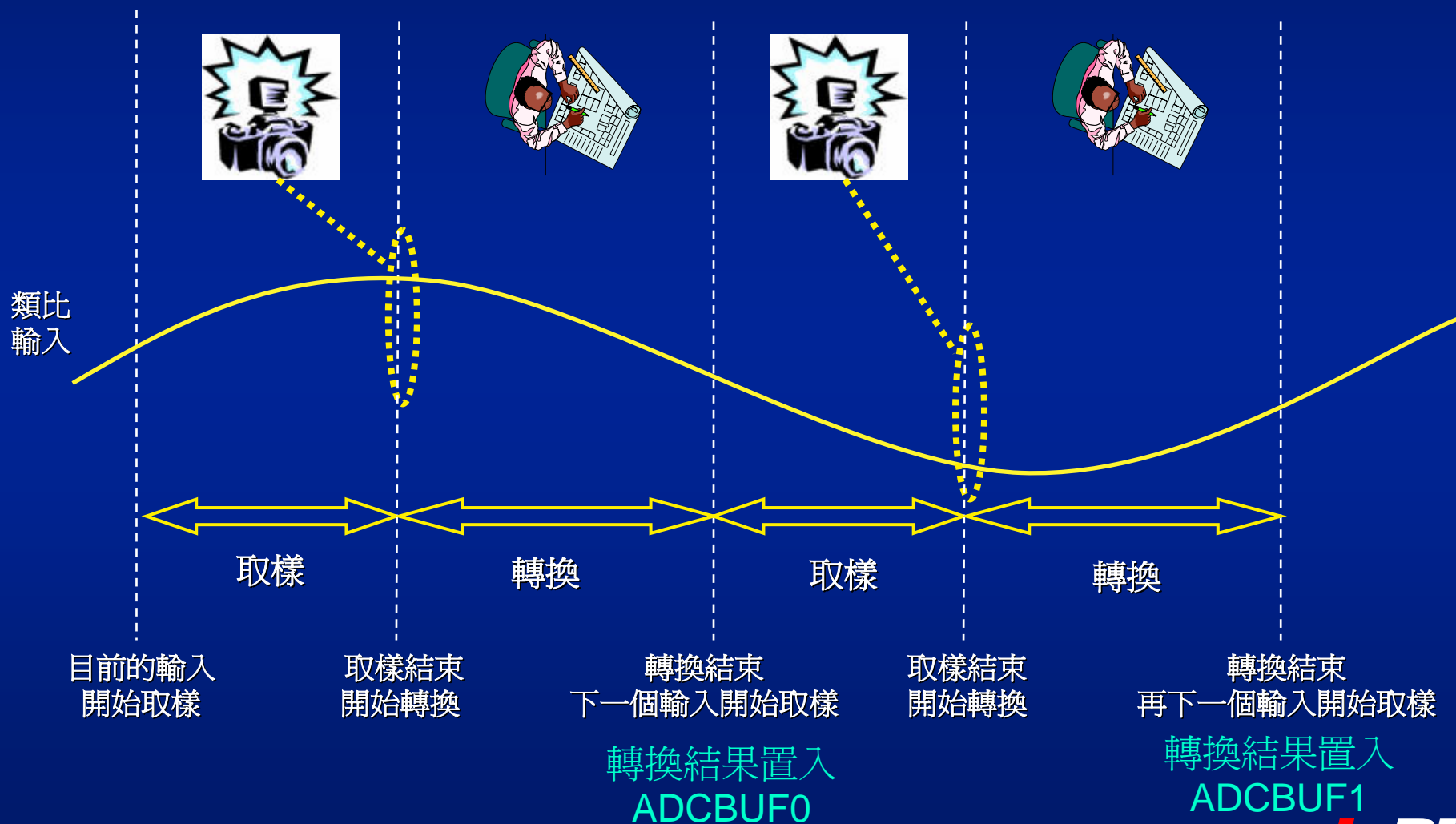
## 手動取樣 / 轉換步驟

條件  $ASAM=0$



# A/D 轉換器

## 多個類比輸入時的轉換方式







MICROCHIP

# A/D 轉換器

## ADCON2 暫存器

定義：

參考電壓

資料輸出緩衝器

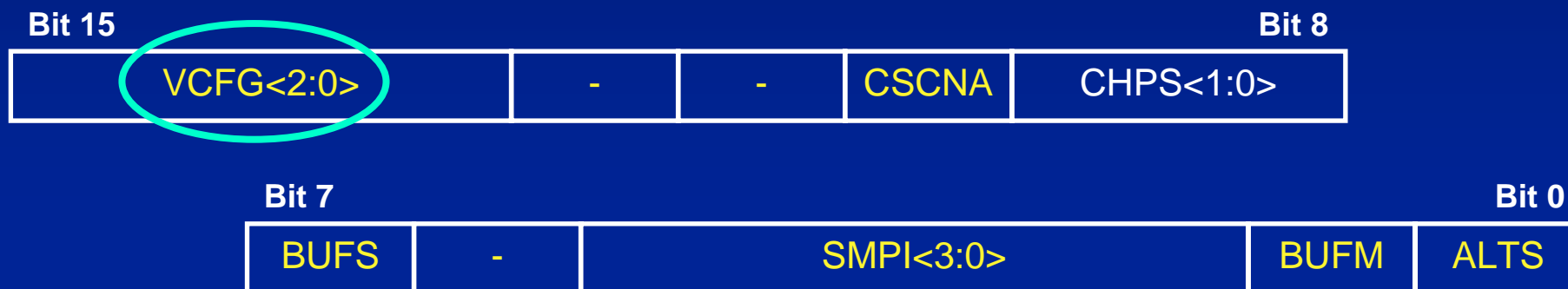
多工器 & 輸入掃描



MICROCHIP

# ADCON2 暫存器

## ADCON2



### VCFG<2:0> - 參考電壓源的選擇

- 000 – AVDD , AVSS
- 001 – 外接 VREF+ , AVSS
- 010 – AVDD , 外接 VREF-
- 011 – 外接 VREF+ , 外接 VREF-
- 1xx – AVDD , AVSS

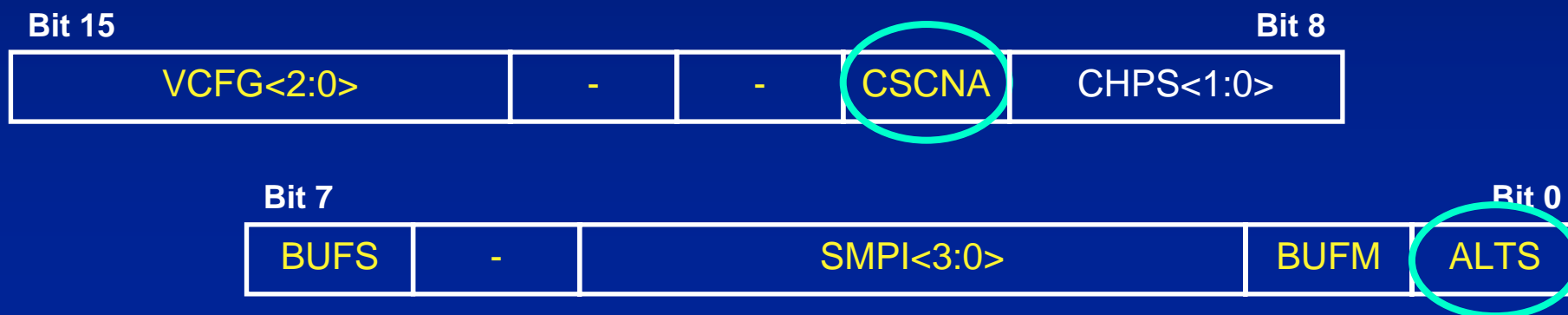
轉換的輸入電壓範圍被限制在參考電壓範圍之間，在此範圍之外(10-bit)的電壓會以最大值(0x3FF) 或以最小值(0x000) 表示



MICROCHIP

# ADCON2 暫存器

## ADCON2



CSCNA : =1時，採用自動掃描方式自A組多工器輸入，B組多工器不具有掃描輸入功能

ALTS : = 1 時，先選 A 組輸入再選 B 組輸入相互交換著選擇多工器 (A→B→A→B→A→B ....)  
= 0 時，輸入只選 A 組多工器

關於此項功能會在後面詳細說明使用方式

**dsPIC**

# ADCON2 暫存器

## ADCON2



**SMPI<3:0>** - 設定 AD 要轉換幾次後才產生一次中斷  
(這些轉換後的資料會被存到 ADCBUF<sub>x</sub> 的暫存器列裡)

**BUFM** – ADCBUF<sub>x</sub> 暫存器列設定成單組或兩組模式

**BUFS** – ADCBUF<sub>x</sub> 採兩組模式時的狀態指示位元

# 轉換結果儲存規則

## ❖ AD 轉換的結果儲存到哪裡？

### ◆ 當 BUFM 位元 = 0

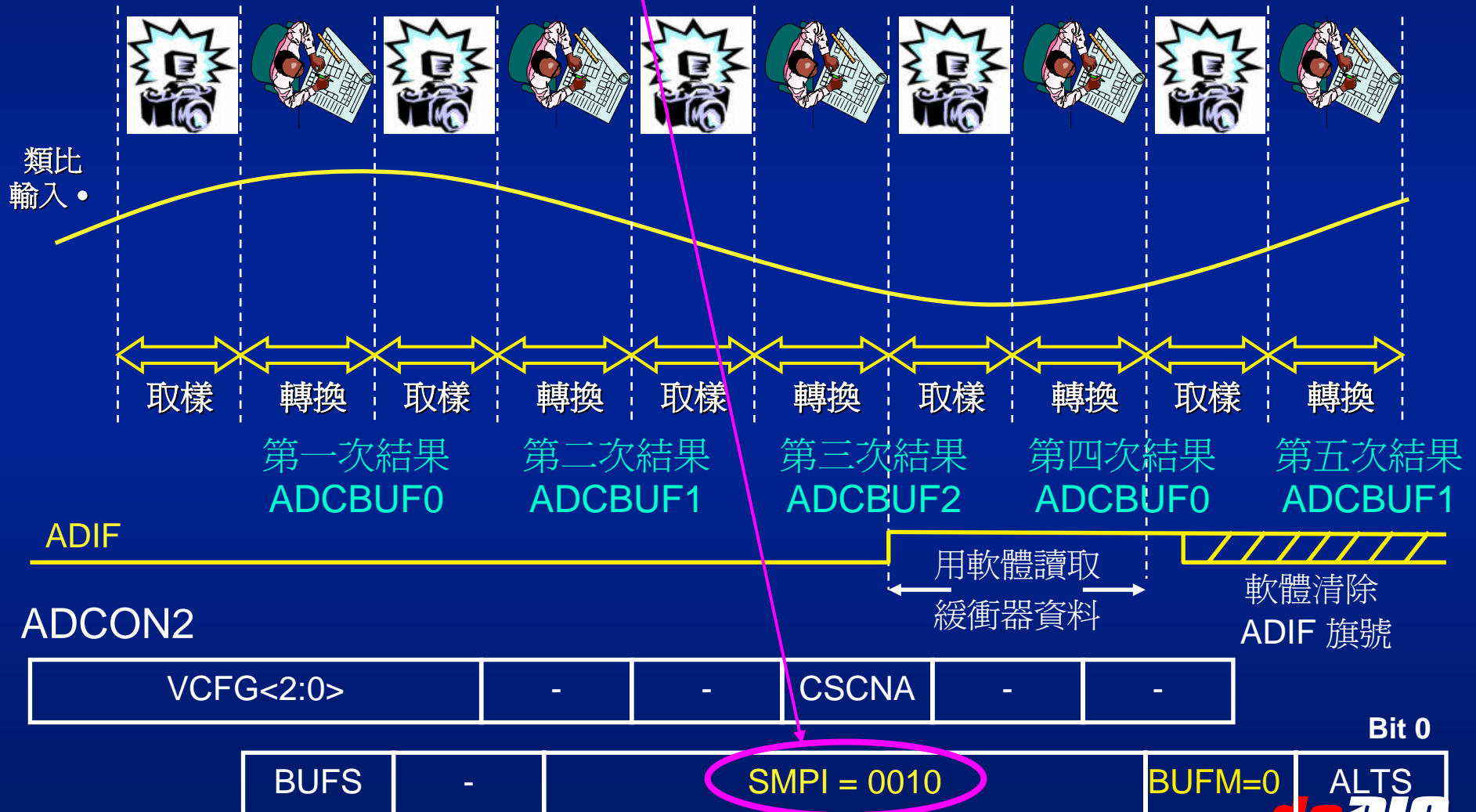
- ◆ 結果存入單一組16個緩衝器裡 - ADCBUF0,1,2...E,F
- ◆ 每次 ADIF 中斷產生後，指標歸零指向 ADCBUF0
- ◆ 每次 ADIF 中斷產生後，新的轉換資料會蓋掉上一次的 存在 ADCBUF0 的資料
- ◆ 考慮中斷發生之後，軟體是否可在資料被下一輪的轉換結果覆蓋前對這一輪的資料做處置！

### ◆ 當 BUFM 位元 = 1

- ◆ 結果會分組自動存入兩組的8個緩衝器裡
- ◆ BUFS 位元會指出目前 AD 轉換使用那組緩衝器
- ◆ BUFS = 1, AD 目前填入第二組緩衝器 ADCBUF8 - F
- ◆ 當使用此模式時，每一輪的轉換最多可在儲存 8 個資料後中斷。
- ◆ 雖然單次可儲存資料量變少，但 CPU 有更多時間間隔可以利用

# A/D 轉換器

## 轉換三次後產生中斷的說明

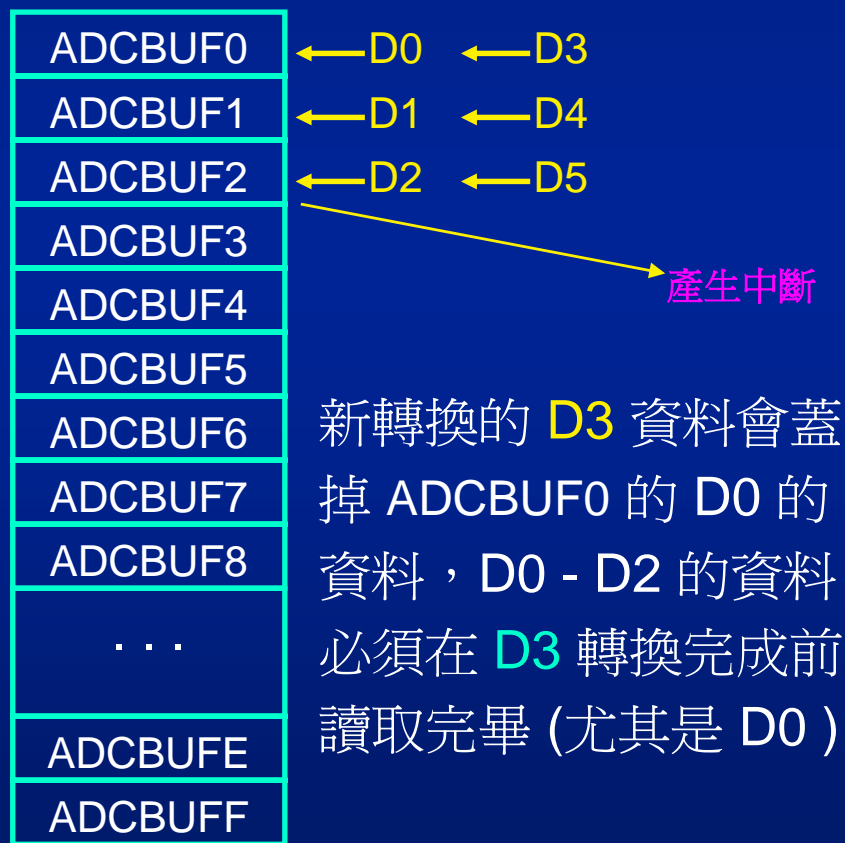


# A/D 轉換器

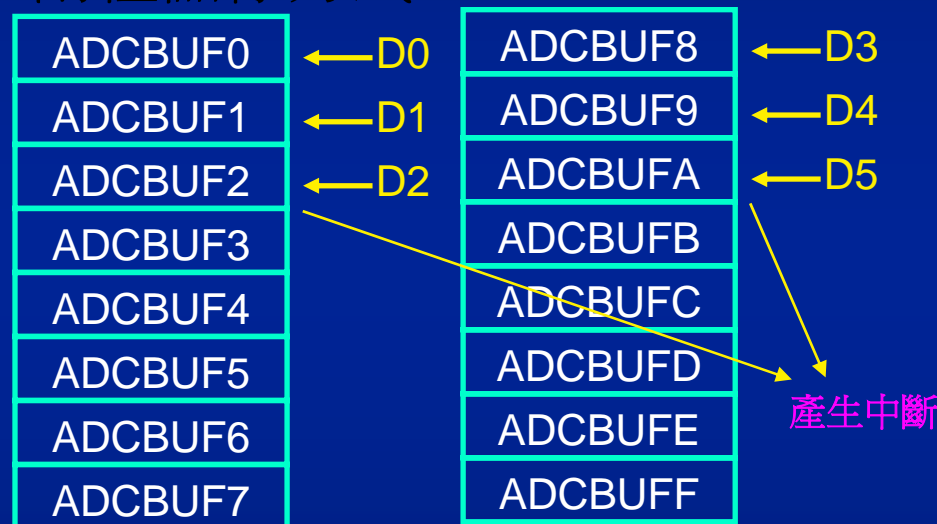
## 轉換儲存模式 & 轉換次數與中斷

設定為三次轉換後產生一次中斷 - **SMPI = 0010**

### 單組儲存方式 - **BUFM = 0**



### 兩組儲存方式 - **BUFM = 1**

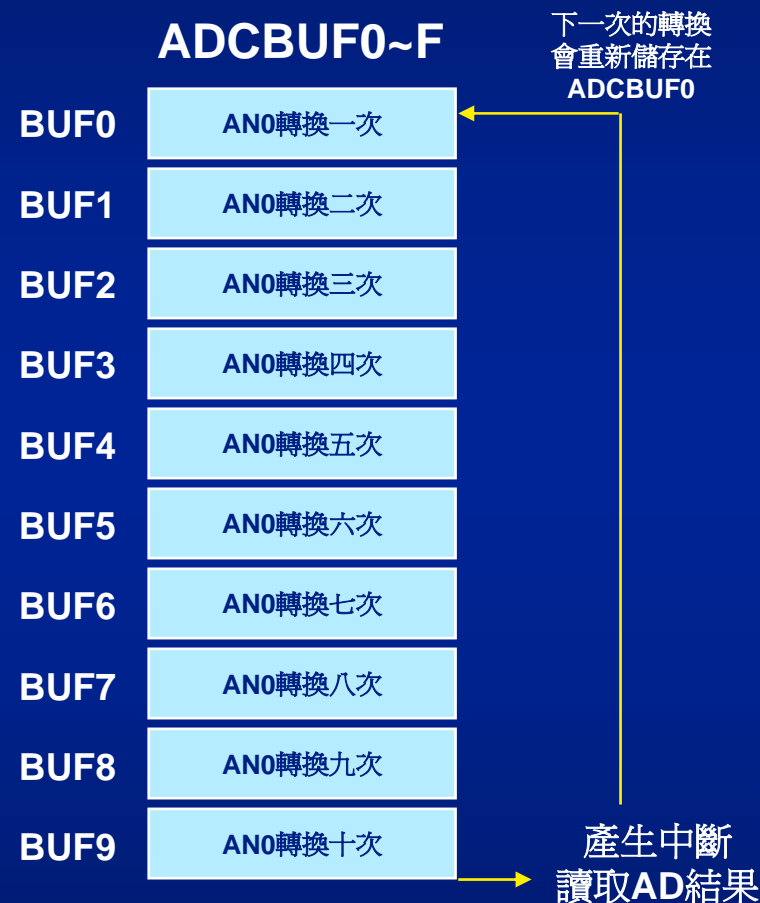


**D6** 的資料會蓋定 ADCBUF0 的 D0，所以 D0 - D2 的資料必須在 **D6** 轉換完成前讀取完畢

# AD 中斷設定

設定需求：

- ANO 為輸入腳，轉換 10 次後產生一次中斷
- ❖  $SMPI<2:0>=1001$ ，轉換10次後中斷一次
- ❖  $BUFM=0$ ，採用單一 buffer (16-word)
- ❖  $ALTS=0$ ，只使用 MUX A 為輸入
- ❖  $CHOSA<3:0>=0000$ ，ANO for CH0+ 輸入
- ❖  $CHONA=0$ ，Vref- for CH0- 輸入
- ❖  $CSCNA=0$ ，輸入掃描禁能(Disable)
- ❖  $CSSL<15:0>= N/A$
- ❖  $CHOSB<3:0>= N/A$
- ❖  $CHONB=N/A$





# 同時取樣的CH選擇

## ADCON2



- ❖ CHPS<1:0> - 選擇同時取樣的輸入數目 (1, 2, 或 4)
  - ◆ 1x = 同時轉換 CH0, CH1, CH2 和 CH3
  - ◆ 01 = 同時轉換 CH0 和 CH1
  - ◆ 00 = 轉換 CH0

# CHPS<1:0> 與 SIMSAM 關係

CHPS<1:0>	SIMSAM	取樣 / 轉換 順序	取樣時間
00	x	取樣 CH0，轉換 CH0	1
01	0	取樣 CH0，轉換 CH0 取樣 CH1，轉換 CH1	2
1X	0	取樣 CH0，轉換 CH0 取樣 CH1，轉換 CH1 取樣 CH2，轉換 CH2 取樣 CH3，轉換 CH3	4
01	1	同時取樣 CH0 CH1 轉換 CH0 轉換 CH1	1
1X	1	同時取樣 CH0, CH1, CH2, CH3 轉換 CH0 轉換 CH1 轉換 CH2 轉換 CH3	1

# 使用同時取樣注意事項 (一)

- ❖ CH0 (以 MUX A 為例)
  - ◆ 正端輸入可從 AN0 ~ AN15 → (CH0SA<3:0>)
  - ◆ 負端輸入有 AN1, Vref- → (CH0NA)
- ❖ CH1 (以 MUX A 為例)
  - ◆ 正端輸入可從 AN0, AN3 → (CH123SA)
  - ◆ 負端輸入有 AN6, AN9, Vref- → (CH123NA<1:0>)
- ❖ CH2 (以 MUX A 為例)
  - ◆ 正端輸入可從 AN1, AN4 → (CH123SA)
  - ◆ 負端輸入有 AN7, AN10, Vref- → (CH123NA<1:0>)
- ❖ CH3 (以 MUX A 為例)
  - ◆ 正端輸入可從 AN2, AN5 → (CH123SA)
  - ◆ 負端輸入有 AN8, AN11, Vref- → (CH123NA<1:0>)

## 使用同時取樣注意事項 (二)

(以使用 MUX A 為例)

- ❖ 若  $SIMSAM=1$  (設定同時取樣模式) &  $CHPS<1:0>=11$  (同時取樣  $CH0\sim CH3$ )
  - ◆  $CH123SA=0$  :
    - ◆ CH1的輸入選 AN0
    - ◆ CH2的輸入選 AN1
    - ◆ CH3的輸入選 AN2
  - ◆  $CH123SA=1$  :
    - ◆ CH1的輸入選 AN3
    - ◆ CH2的輸入選 AN4
    - ◆ CH3的輸入選 AN5

只有兩種選擇  
此項規格在電路設計時要特別留意



MICROCHIP

# A/D 轉換器

## ADCON3 暫存器

定義：

自動取樣時間

轉換時脈設定



MICROCHIP

# ADCON3 暫存器

## ADCON3



SAMC<4:0> : 設定自動轉換時的取樣時間 ( $0 T_{AD}$  to  $31 T_{AD}$ )

ADRC : =1時，使用內建RC振盪時脈；=0 時，使用系統時脈

ADCS<5:0> - AD 轉換時脈的時間 ( $T_{AD}$ )

$$111111 = (T_{cy} / 2) (ADCS<5:0> + 1) = 32 T_{cy}$$

.....

$$000000 = T_{cy}/2$$

# 自動取樣轉換的時間計算

## ➤ 必需先設定為自動取樣與轉換模式

- 假設石英晶體用7.3728MHz，並使用16 倍的倍頻電路使工作頻率為 117.9648MHz = 8.477nS (Fosc)
- $T_{cy} = (1 / F_{osc}) \times 4 = 33.91nS$

## ➤ 在規格書裡有規定

- 最小的 TAD 需大於 167nS，最小的 轉換時間 =  $167nS \times 12 = 2\mu S$
- $T_{ad} = (ADCS<5:0>+1) T_{cy} / 2$
- 若  $ADCON3 = 0x0709$  ( $ADCS=001001$ ,  $SAMC=01111$ )

$$T_{ad} = (ADCS<5:0>+1) T_{cy} / 2 = (9+1) T_{cy} / 2 = T_{cy} \times 5 = 33.91nS \times 5 = 169nS,$$

轉換時間  $T_{conv} = 12T_{ad}$ ，取樣時間 =  $15 T_{ad}$

$$A/D = \text{取樣時間} + \text{轉換時間} = 15 T_{ad} + 12 T_{ad} = (15 + 12) \times 169\mu S = 4.56\mu S$$

# 轉換時間計算範例

## 10-bit ADC

最小的 TAD = 167ns

類比  
輸入

範例：

$F_{osc} = 29.4912\text{MHz}$

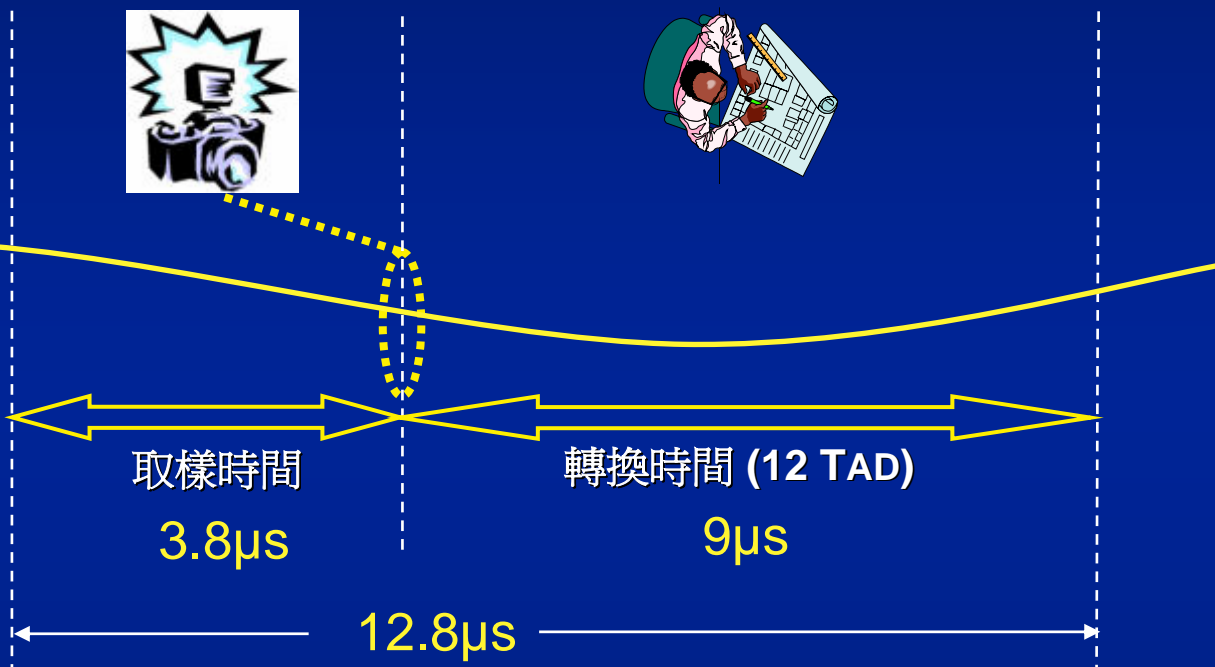
$TCY = 136\text{ns}$

$TAD = TCY(ADCS+1)/2 = 748\text{ns}$  ( $T_{conv.} = 12 TAD = 9 \mu\text{s}$ )

$TSAMP = TAD \times SAMC = 748\text{ns} \times 5 = 3.8\mu\text{s}$

ADCON3

-	-	-	SAMC = 00101	ADRC = 0	-	ADCS = 001010
---	---	---	--------------	----------	---	---------------





# 輸入阻抗與取樣時間

➤ 觀念！輸入阻抗越大取樣時間越長

- 公式： $T_{SMP} = \text{放大器穩定時間 (Tamp)} + \text{取樣電容充電時間(Tc)} + \text{溫度係素 (T_{COFF})}$  – 參考 dsPIC30F Family Reference Manual
  - Tamp 為固定時間 = 0.5uS
  - $T_c = -C_{HOLD} (R_{IC} + R_{SS} + R_{in}) \ln (1/2n) \text{Secs}$
  - $T_{COFF} = (\text{Temp} - 25^\circ\text{C}) (0.05\mu\text{S} / ^\circ\text{C})$  溫度越高取樣時間變長

條件：

1.  $C_{HOLD} = 18\text{Pf}$
2.  $R_{in} = 2.5\text{Kohm}$
3.  $V_i = 4095 \text{ LSB}$  ,  $n=4096$  for full scale input voltage
4.  $R_{ss} = 1.2\text{Kohm}$
5.  $\text{Temp} = 25^\circ\text{C}$

$$T_c = -18\text{pF} (250 + 1.2\text{K} + 2.5\text{K}) \ln (1/8192) \\ = 0.641\mu\text{S}$$

$$T_{SMP} = 0.5\mu\text{s} + 0.641\mu\text{S} + (25^\circ\text{C} - 25^\circ\text{C})(0.05/^\circ\text{C}) \\ = \mathbf{1.141\mu\text{S}} \text{ (取樣時間)}$$



MICROCHIP

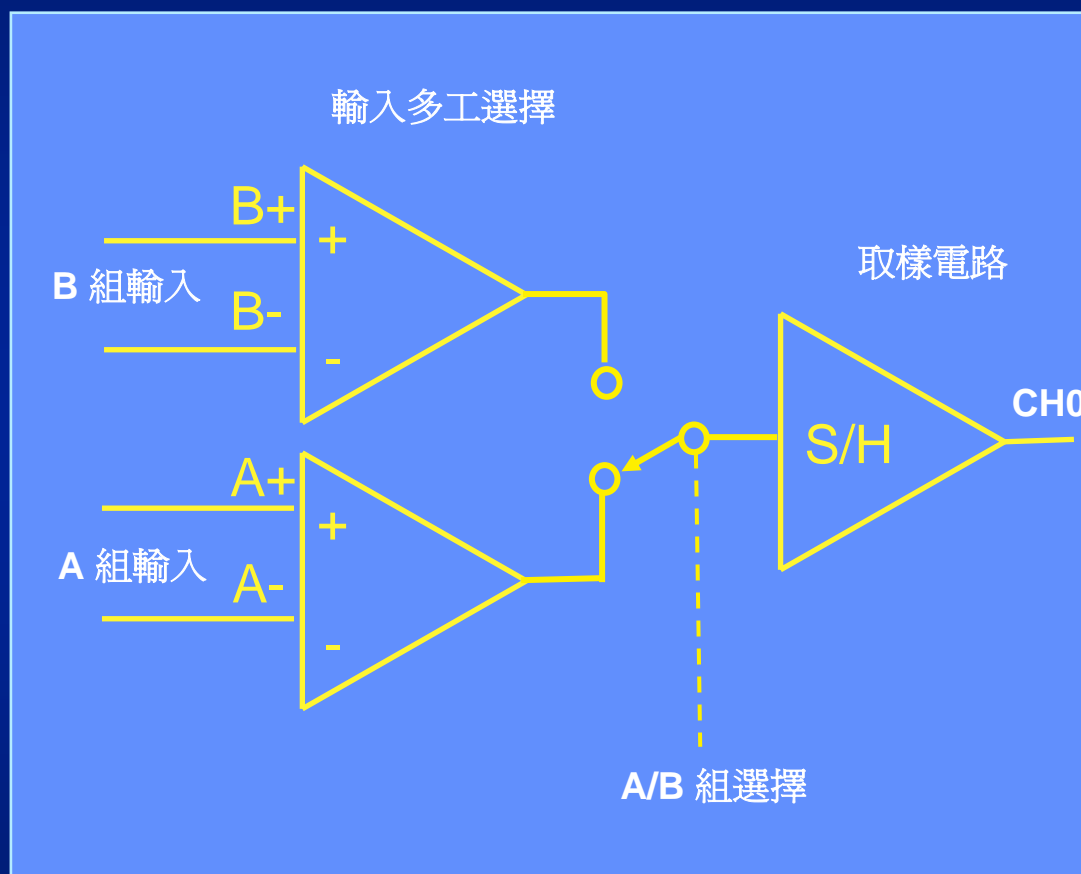
# A/D 轉換器 ADCHS 暫存器

定義：

A、B 兩組多工輸入選擇

# 輸入多工器

- ❖ 10-bit AD 轉換器有兩組多工輸入選擇(A及B)
- ❖ 每次的轉換僅能使用一組輸入
- ❖ 轉換的電壓值為正輸入端電壓減去負輸入端電壓



輸出值為：
$$[(V_{A+} - V_{A-}) - V_{ref-}] / (V_{ref+} - V_{ref-}) * 1024 - 1$$

# ADCHS 暫存器 MUX-A 的輸入選擇

ADCHS



CH0NA : A 組多工器負端輸入選擇

1 = 輸入選 AN1 , 0 = 輸入選 Vref-

CH0SA<3:0> : A 組多工器正端輸入選擇

1111 : 選擇 AN15為輸入端

.....

0000 : 選擇 AN0為輸入端

# ADCHS 暫存器 MUX-B 的輸入選擇

ADCHS



**CH0NB : B 組多工器負端輸入選擇**

1 = 輸入選 AN1 , 0 = 輸入選 Vref-

**CH0SB<3:0> : B 組多工器正端輸入選擇**

1111 : 選擇 AN15為輸入端

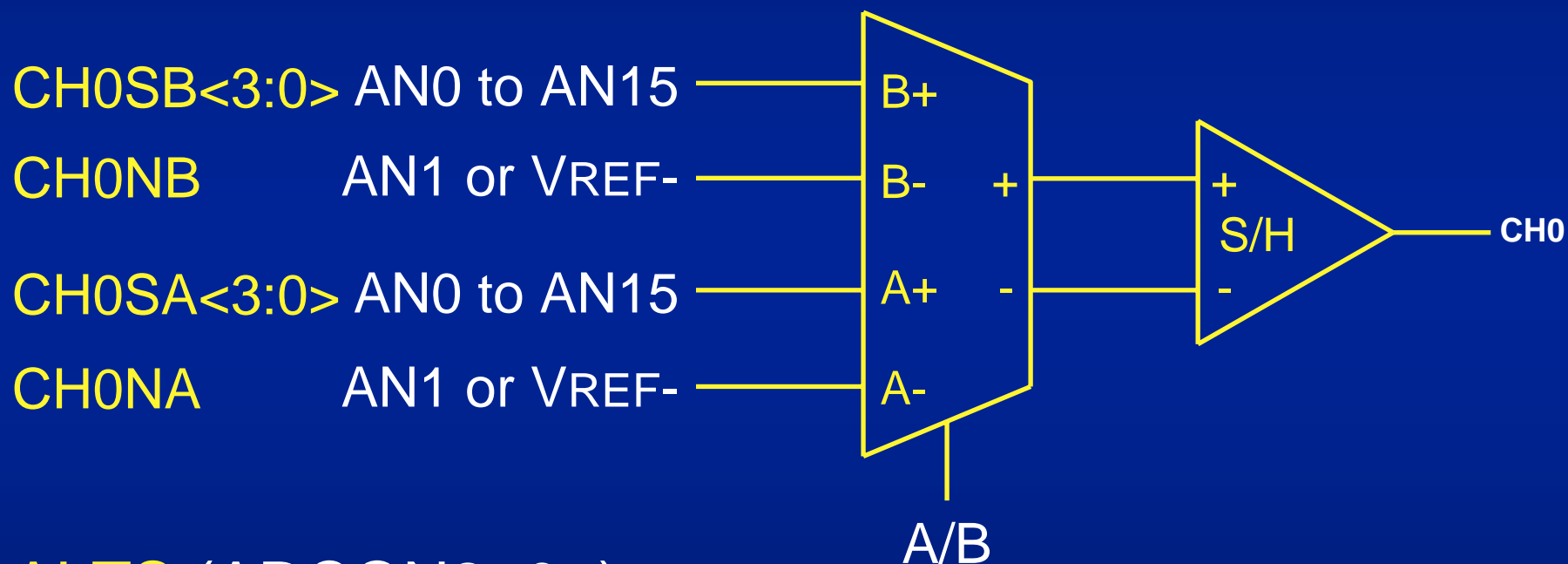
.....

0000 : 選擇 AN0為輸入端

# A/D 轉換器

## 多工 (交錯式) 輸入

多工器的取樣輸出 CH0



**ALTS (ADCON2<0>)**

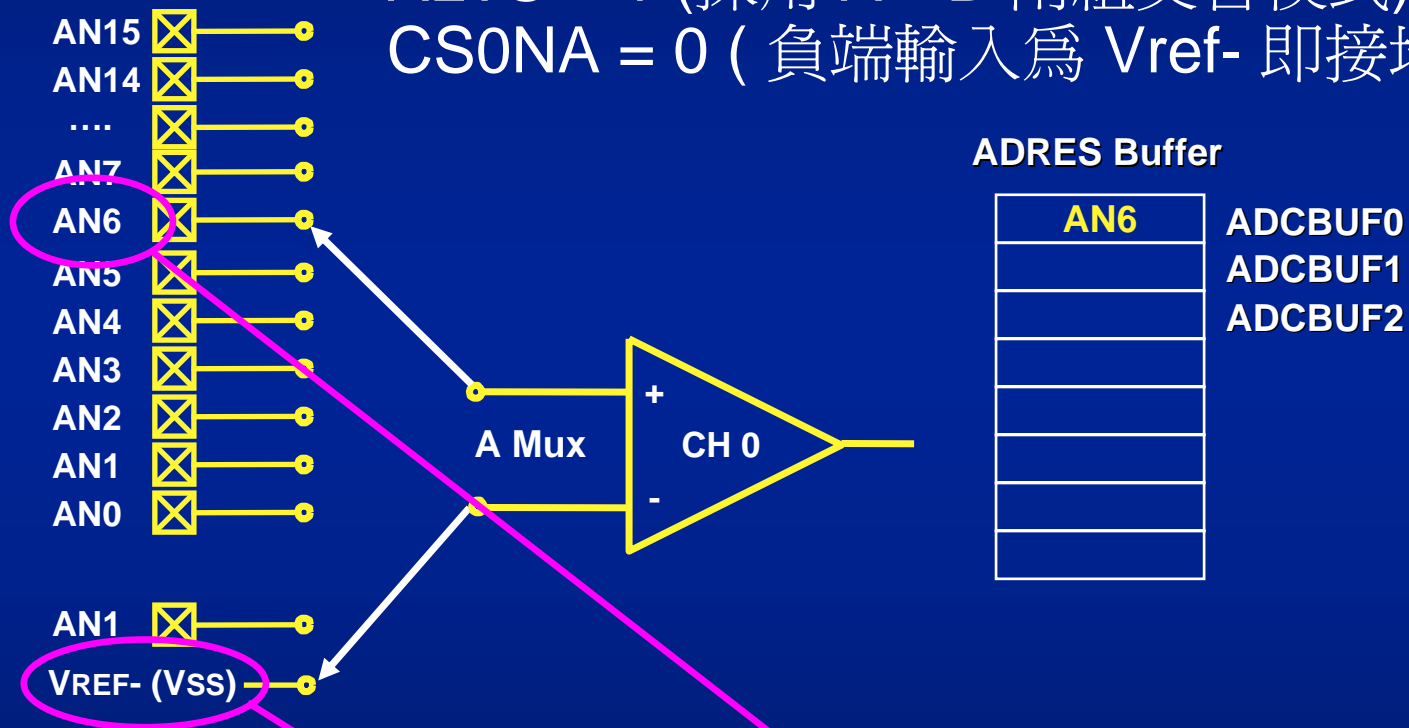
= 0時，輸入只選 A 組多工器

= 1時，先選 A 組輸入再選 B 組輸入相互交替

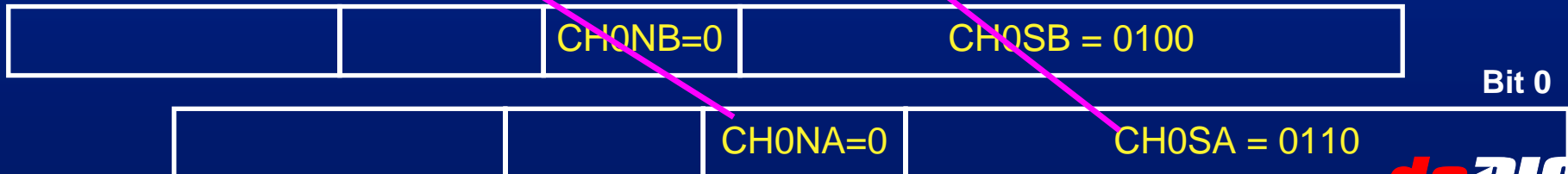
# A/D 轉換器

## 多工 (交錯式) 輸入說明 (一)

ALTS = 1 (採用 A、B 兩組交替模式)  
 CS0NA = 0 (負端輸入為 Vref- 即接地)

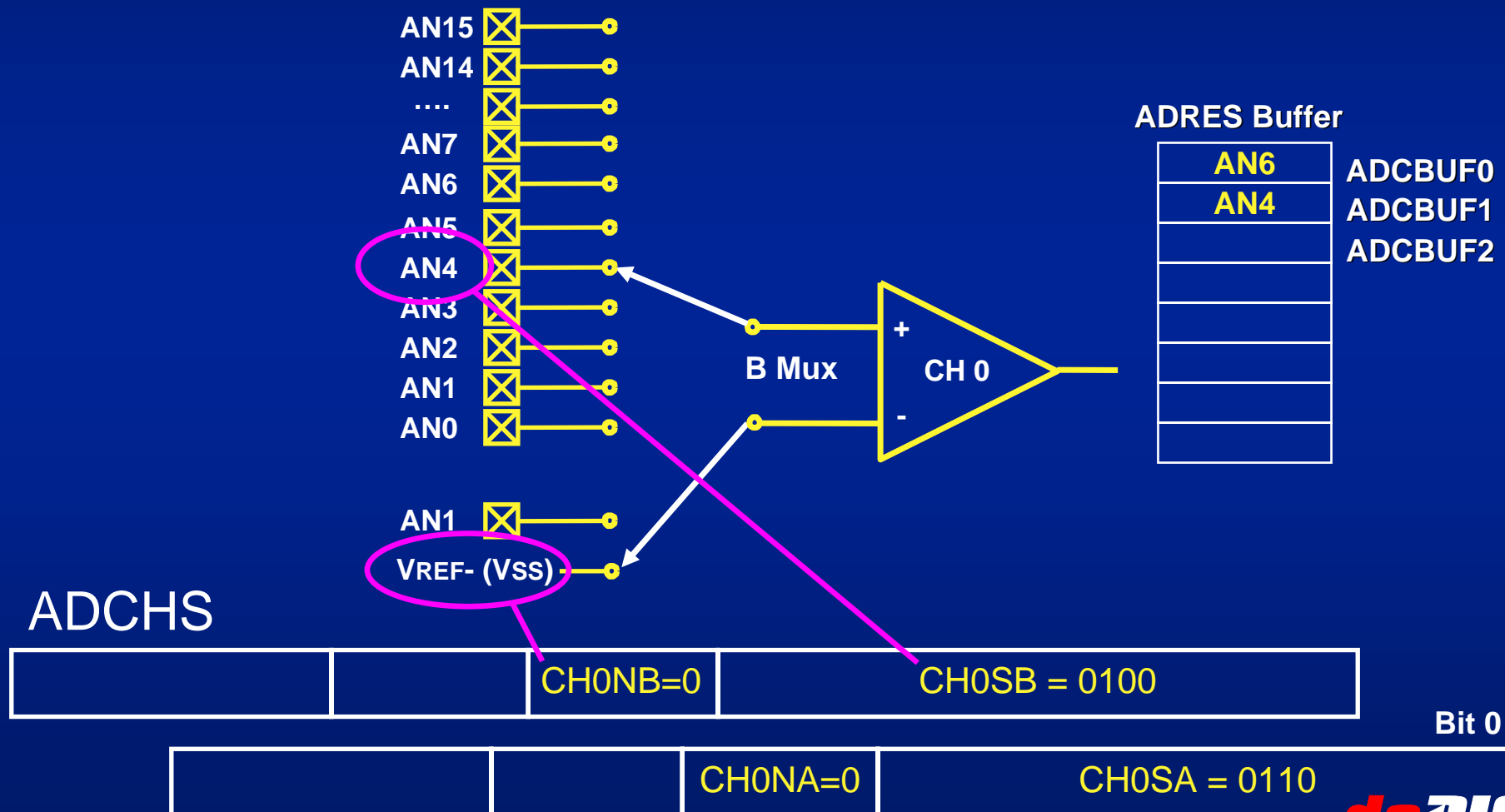


ADCHS



# A/D 轉換器

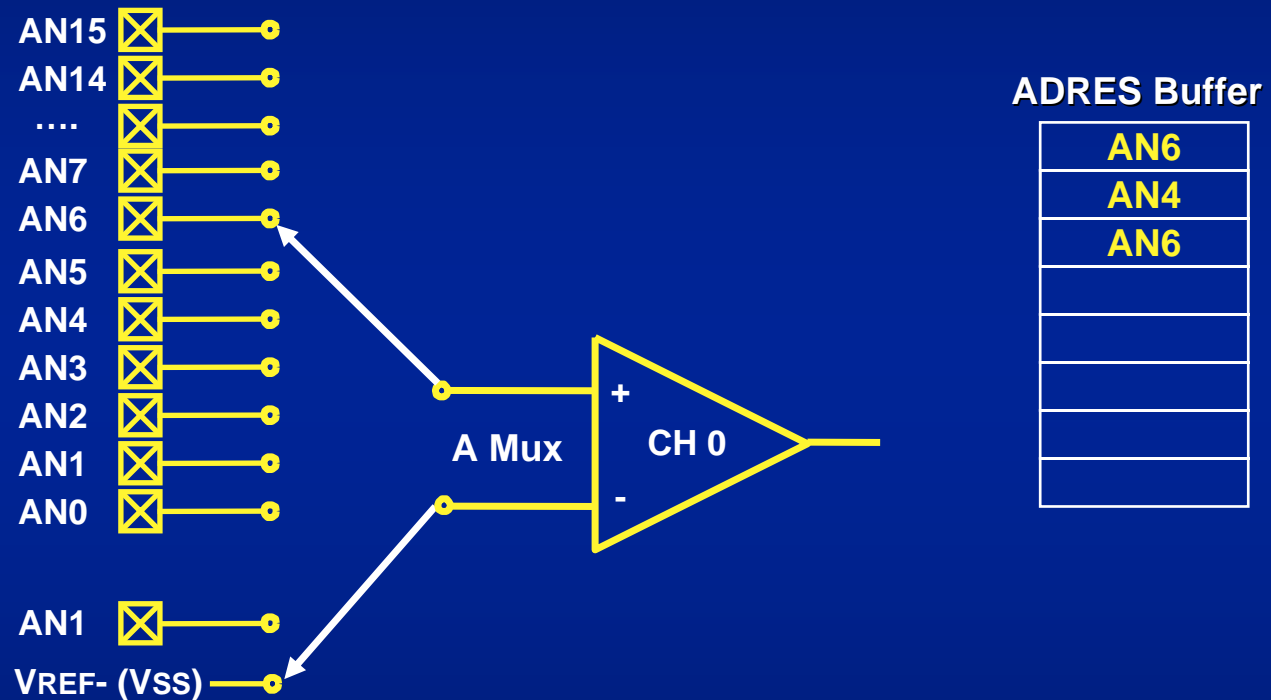
## 多工 (交錯式) 輸入說明 (二)





# A/D 轉換器

## 多工 (交錯式) 輸入說明 (三)



ADCHS

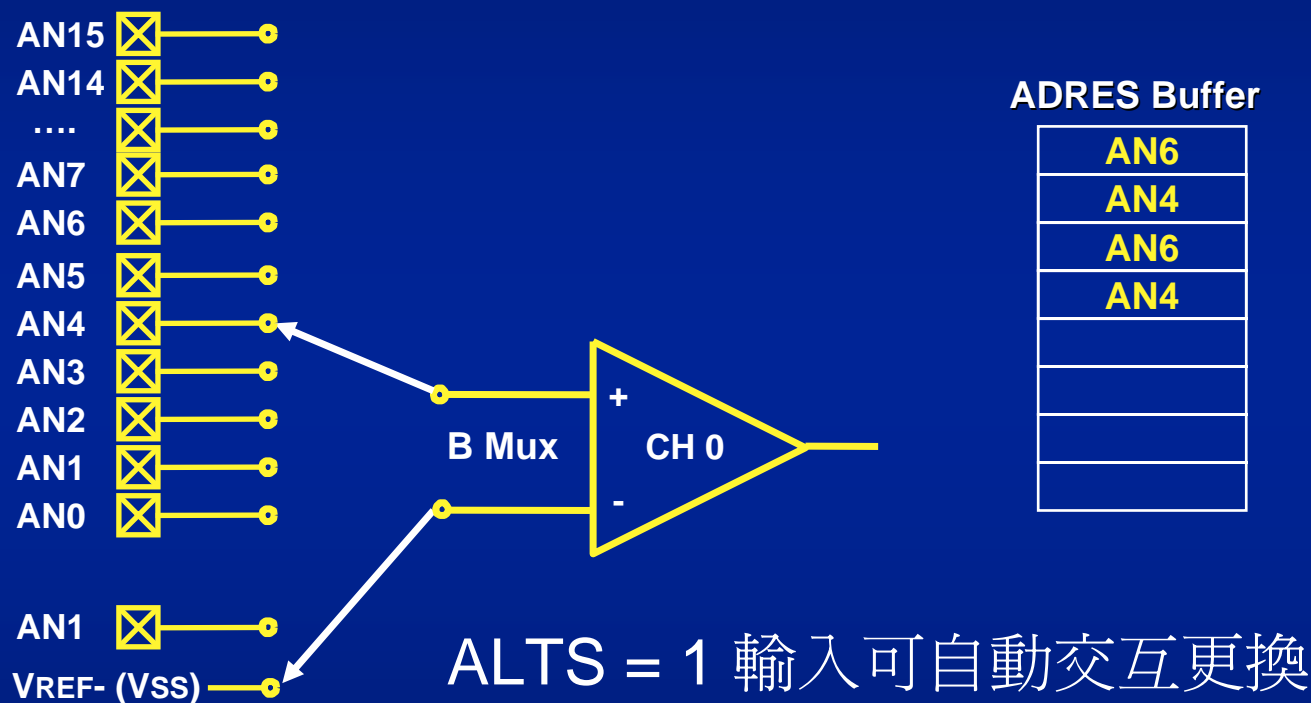
		CH0NB=0	CH0SB = 0100
--	--	---------	--------------

		CH0NA=0	CH0SA = 0110
--	--	---------	--------------

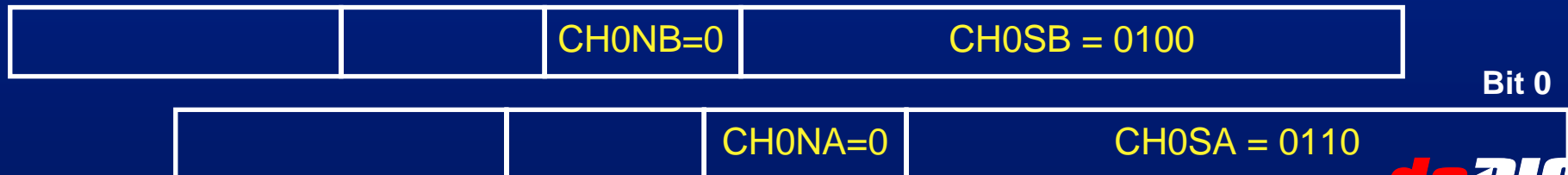
Bit 0

# A/D 轉換器

## 多工 (交錯式) 輸入說明 (四)



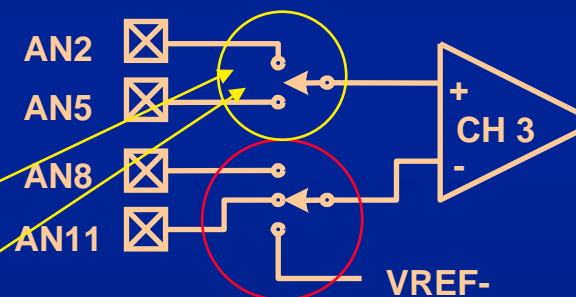
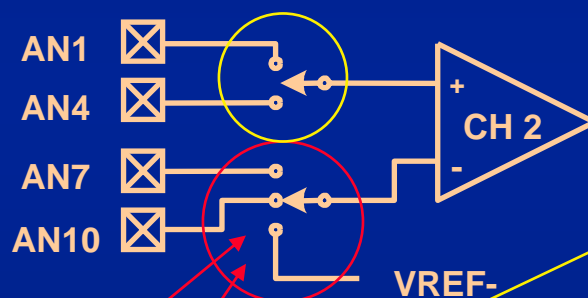
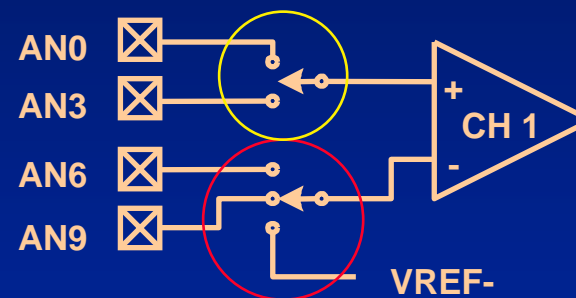
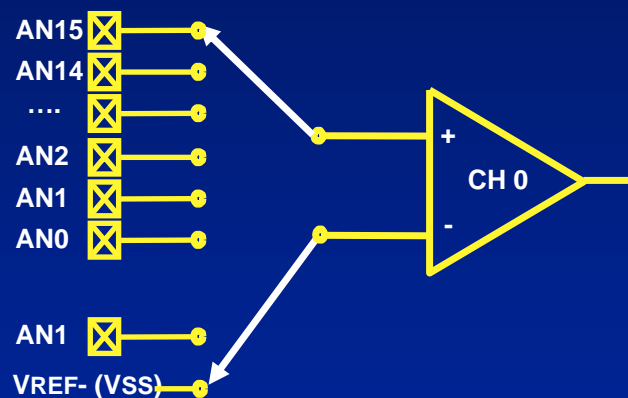
ADCHS





MICROCHIP

# 10-bit A/D 轉換器



ADCHS Register





MICROCHIP

# A/D 轉換器

## ADCSSL 暫存器

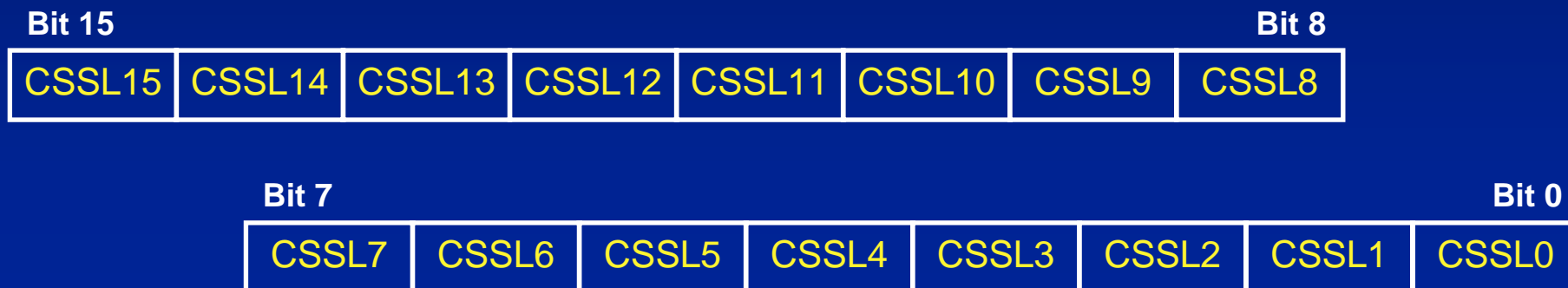
定義：  
類比輸入腳位掃描選擇



MICROCHIP

# ADCSSL 暫存器

## ADCSSL



CSSL<15:0> - 輸入掃描選擇

- 1 – 啟動掃描該腳位
- 0 – 不掃描

啟動掃描功能需將 **CSCNA (ADCON2<10>)** 位元設為 1

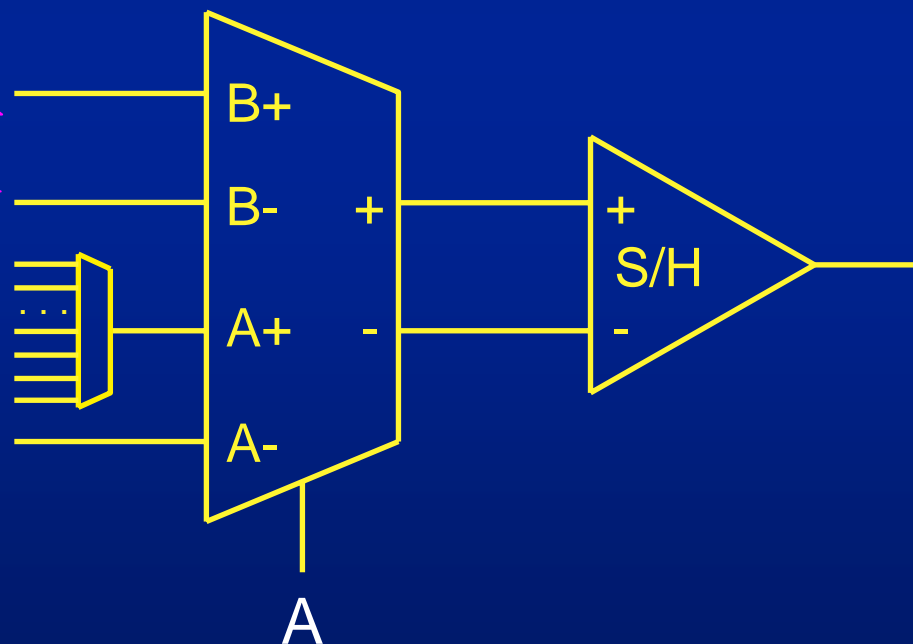
# A/D 轉換器 - 掃描輸入

- ❖ 注意！只有 A 多工器的輸入端才有掃描之功能
- ❖ 這時  $CH0SA<3:0>$  的輸入被  $CSSL<15:0>$  的輸入取代
- ❖ 啟動掃描功能時，也可以將  $ALTS$  設為 1 以啟動 A / B 交互切換功能

範例先討論  $ALTS=0$  的例子  
即暫不使用 B 組多工器輸入

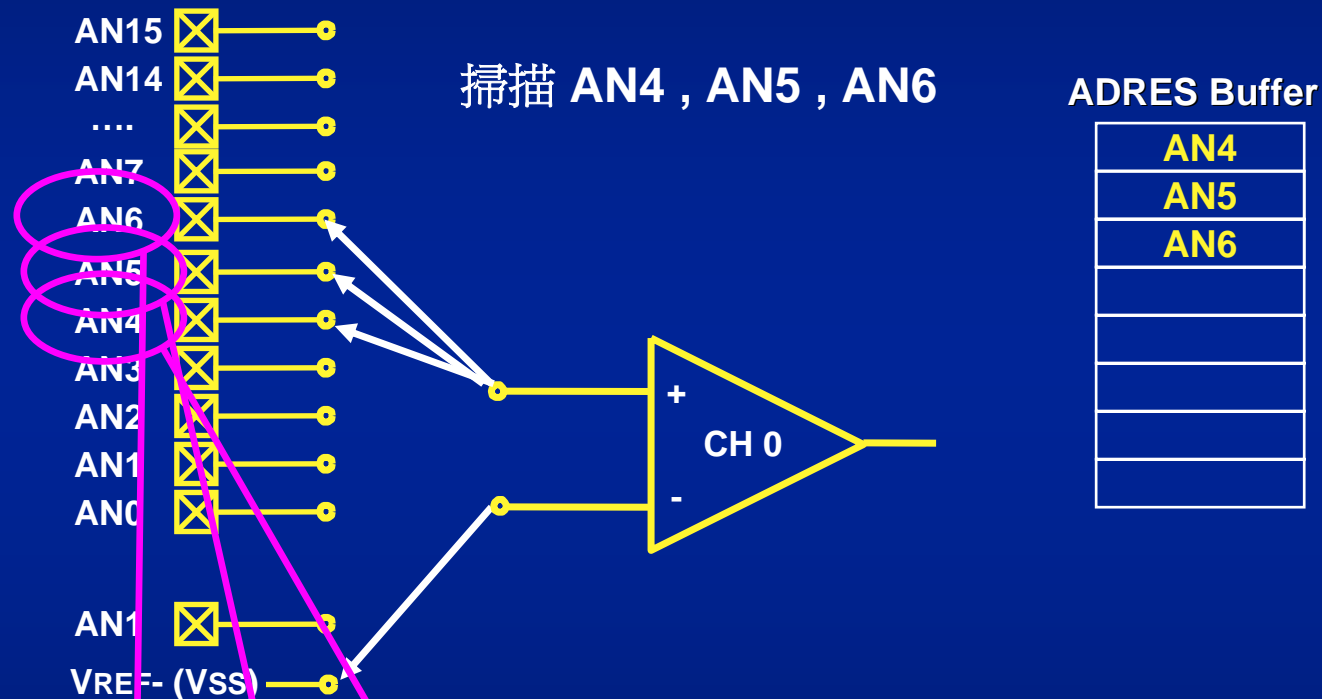
$CSSL<15:0>$  AN0 to AN15

$CH0NA$  AN1 or  $V_{REF-}$



# A/D 轉換器 掃描輸入說明

ALTS = 0  
CSCNA = 1



ADCSSL

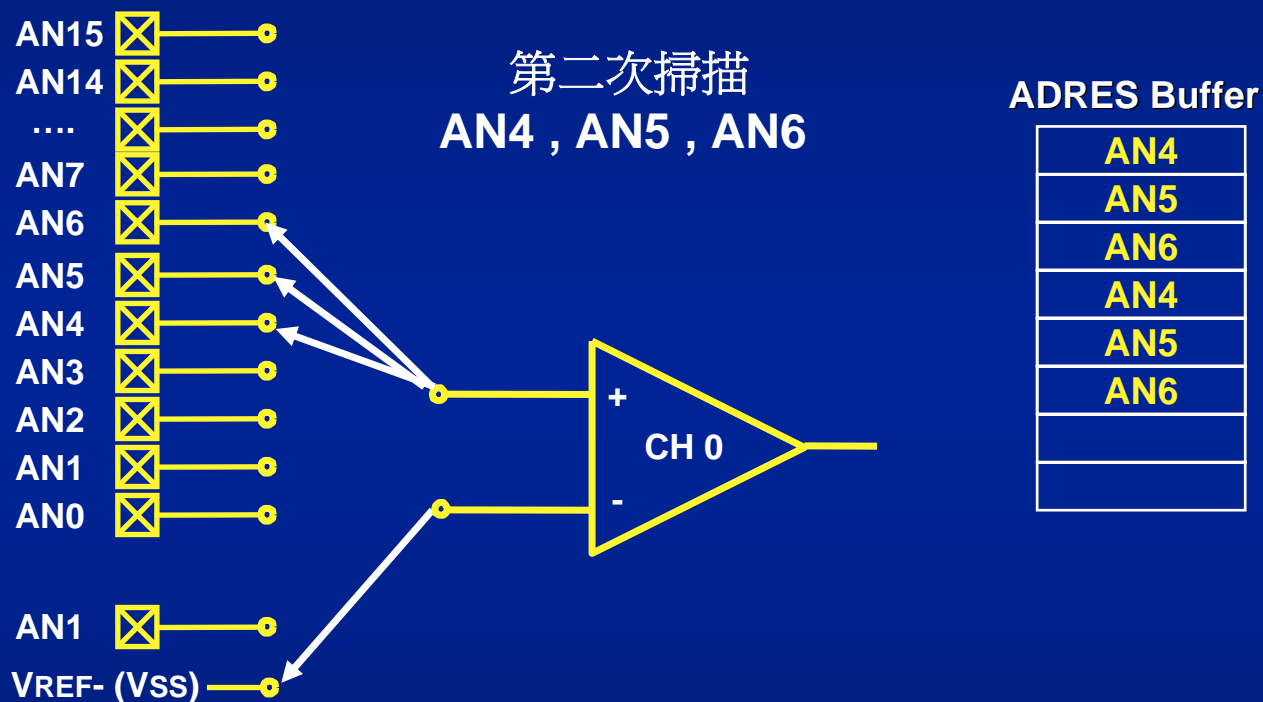
CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8
--------	--------	--------	--------	--------	--------	-------	-------

CSSL7	CSSL6=1	CSSL5=1	CSSL4=1	CSSL3	CSSL2	CSSL1	CSSL0
-------	---------	---------	---------	-------	-------	-------	-------

Bit 0

# A/D 轉換器 掃描輸入說明

ALTS = 0  
CSCNA = 1



## ADCSSL

CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8
--------	--------	--------	--------	--------	--------	-------	-------

CSSL7	CSSL6=1	CSSL5=1	CSSL4=1	CSSL3	CSSL2	CSSL1	CSSL0
-------	---------	---------	---------	-------	-------	-------	-------

Bit 0



# A/D轉換器

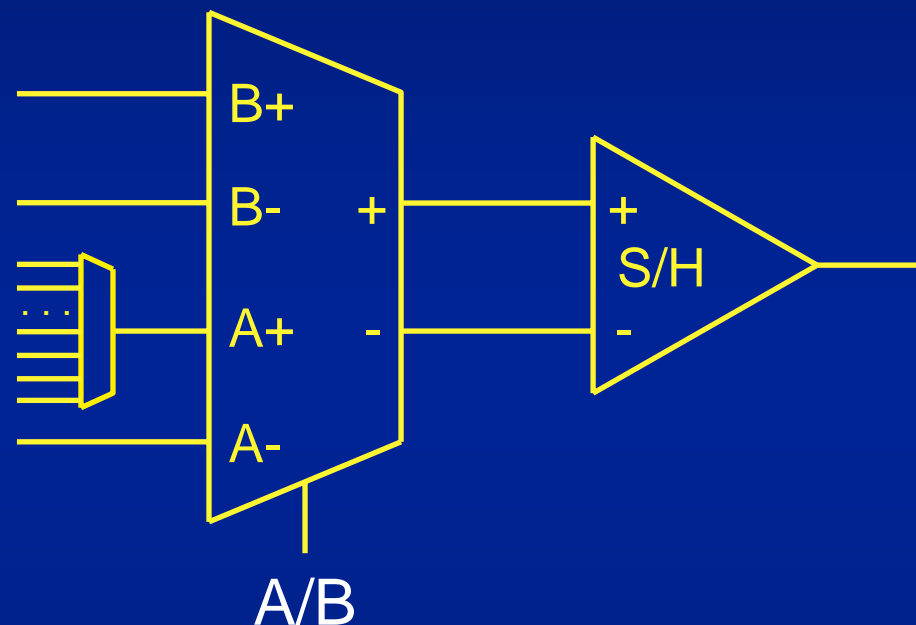
## 使用多工器與掃描輸入說明

CH0SB<3:0> AN0 to AN15

CH0NB AN1 or VREF-

CSSL<15:0> AN0 to AN15

CH0NA AN1 or VREF-



**ALTS** – 從 A 組輸入先掃描一個輸入再選 B 組輸入，  
A / B 兩組之間相互交替做 AD 轉換

**CSCNA** – 啟動掃描多工器 A 的輸入功能



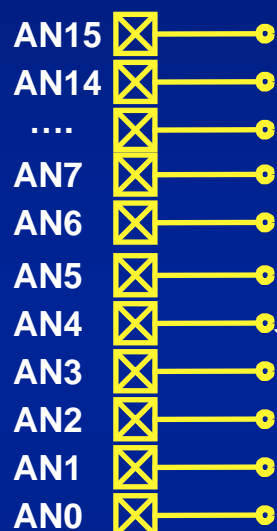
MICROCHIP

ALTS = 1  
CSCNA = 1

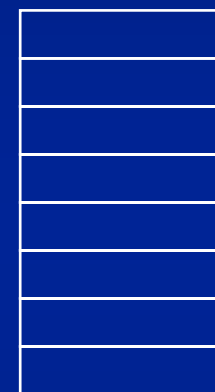
多工器 A  
的輸入

# A/D轉換器

## 使用多工器與掃描輸入說明



ADRES Buffer



ADCHS





MICROCHIP

ALTS = 1  
CSCNA = 1

多工器 A  
的輸入

# A/D轉換器

## 使用多工器與掃描輸入說明



A Mux

CH 0

ADRES Buffer

AN4

ADCSSL

CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8
--------	--------	--------	--------	--------	--------	-------	-------

CSSL7	CSSL6	CSSL5=1	CSSL4=1	CSSL3	CSSL2	CSSL1	CSSL0
-------	-------	---------	---------	-------	-------	-------	-------

Bit 0



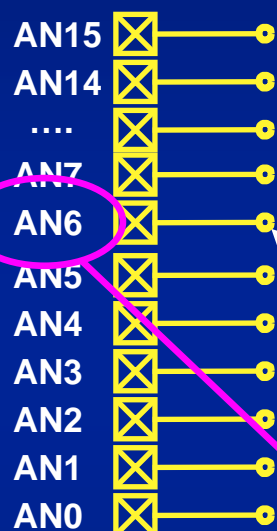
MICROCHIP

ALTS = 1  
CSCNA = 1

多工器 B  
的輸入

# A/D轉換器

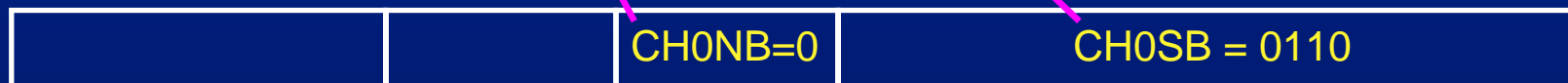
## 使用多工器與掃描輸入說明



ADRES Buffer

AN4
AN6

ADCHS



Bit 0



MICROCHIP

ALTS = 1  
CSCNA = 1

多工器 A  
的輸入

ADCSSL

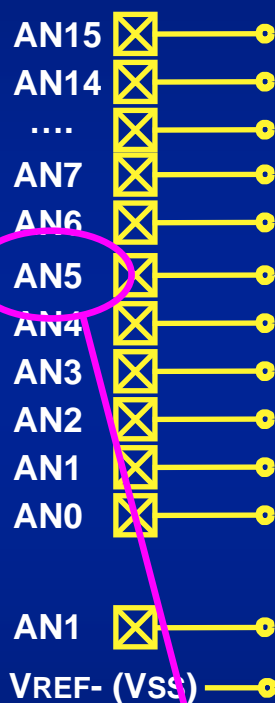
CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8
--------	--------	--------	--------	--------	--------	-------	-------

CSSL7	CSSL6	CSSL5=1	CSSL4=1	CSSL3	CSSL2	CSSL1	CSSL0
-------	-------	---------	---------	-------	-------	-------	-------

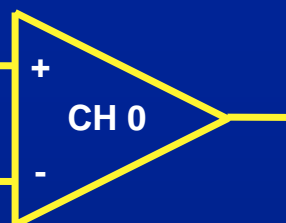
Bit 0

# A/D轉換器

## 使用多工器與掃描輸入說明



A Mux



ADRES Buffer

AN4
AN6
AN5



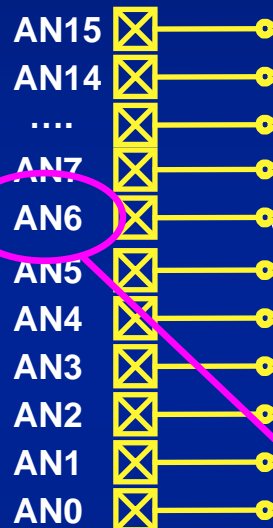
MICROCHIP

ALTS = 1  
CSCNA = 1

多工器 B  
的輸入

# A/D轉換器

## 使用多工器與掃描輸入說明



ADRES Buffer

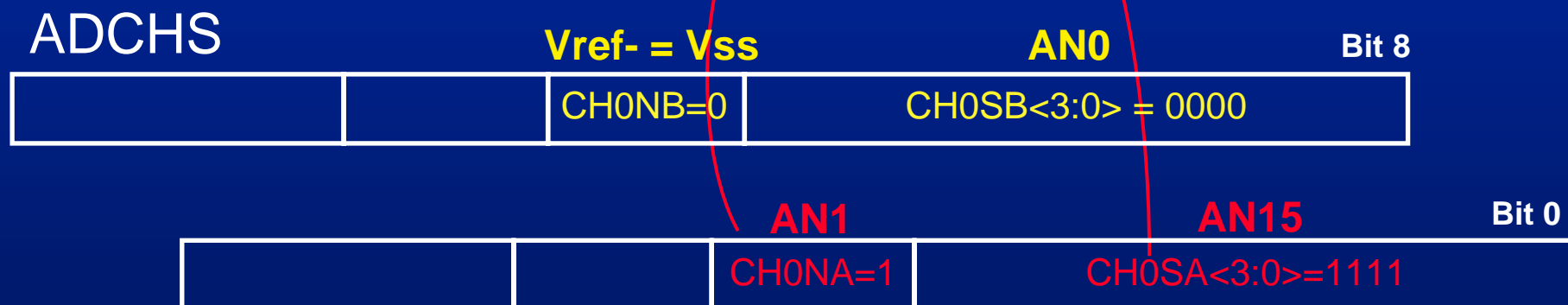
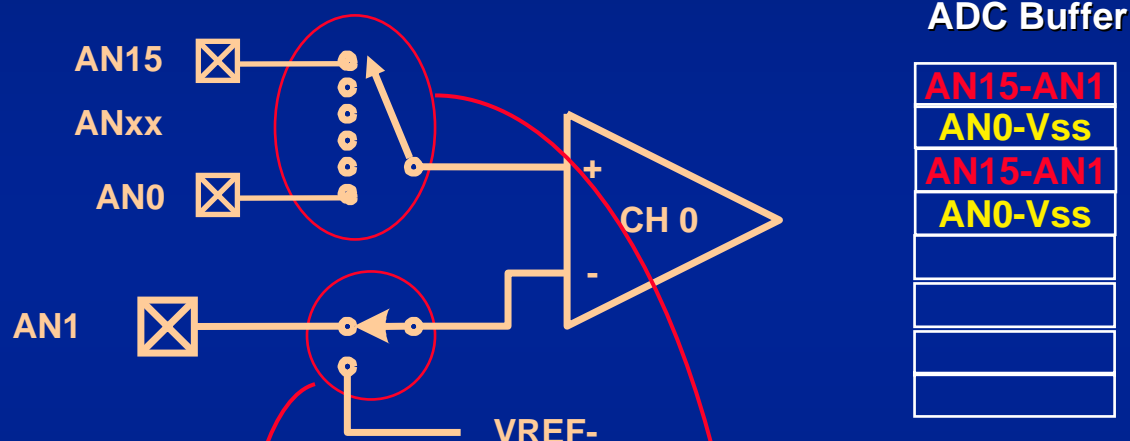
AN4
AN6
AN5
AN6

ADCHS



# 差動輸入與交互掃描

- ❖ 輸入的設計是採用差動式輸入，正端的輸入可以減去負端的 AN1 或 VREF- 的電壓。



# 10-bit A/D 設定步驟

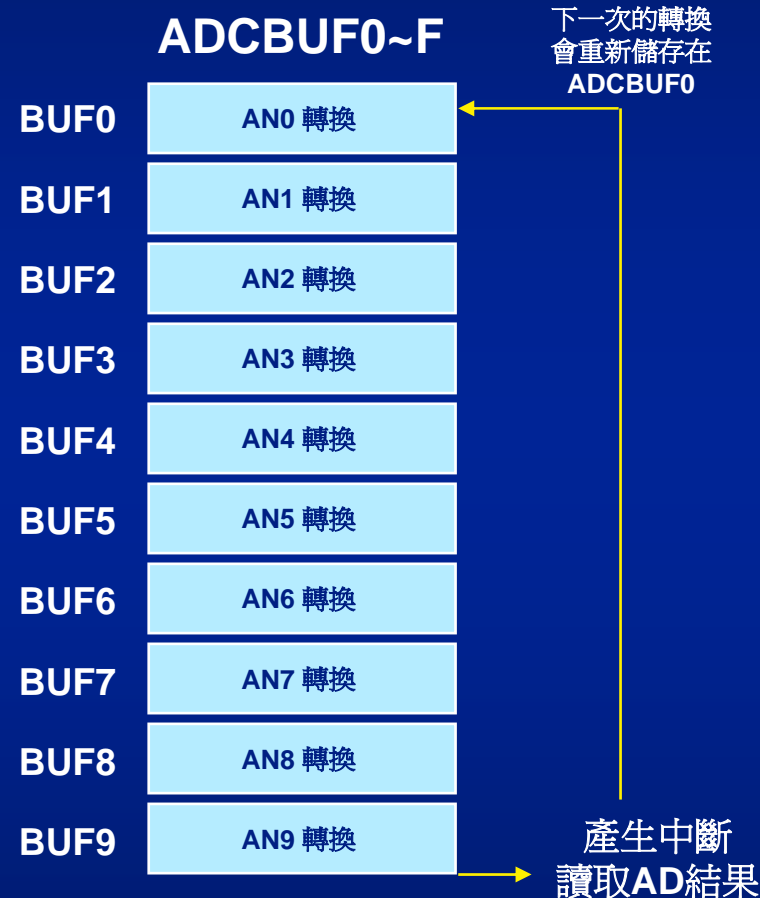
- ❖ 選擇 / 設定類比輸入腳 (ADPCFG<15:0>)
- ❖ 設定參考電壓源 (ADCON2<15:12>)
- ❖ 選擇資料輸出格式 (ADCON1<9:8>)
- ❖ 設定AD轉換觸發模式 (ADCON1<7:5>)
- ❖ 設定轉換時脈的時間 (ADCON3<5:0>)
- ❖ 如需設定自動取樣 (使用自動觸發模式)
  - ◆ 設定自動取樣(ADCON3<12:8>)
- ❖ 選擇 A/D 輸入模式及多工群組 (ADCON2<ALTS,CSCNA>)
  - ◆ 設定 ALTS , 如需使用多工群組 A 及群組 B
  - ◆ 是否需要輸入掃描功能
- ❖ 是否使用同時取樣模式
  - ◆ (ADCON1<SIMSAM>), (ADCON2<9:8>)
- ❖ 設定中斷模式 (ADCON2<5:2>, 啓動中斷, 中斷優先權設定)
- ❖ 啓動 AD 模組



# AD 掃描輸入與中斷設定

設定需求：

- ANO-AN9 為輸入腳，轉換 10 次後產生一次中斷
- ❖  $SMPI<2:0>=1001$ ，轉換10次後中斷一次
- ❖  $BUFM=0$ ，採用單一 buffer (16-word)
- ❖  $ALTS=0$ ，只使用 MUX A 為輸入
- ❖  $CHOSA<3:0>=N/A$ (掃描模式下該輸入無效)
- ❖  $CHONA=0$ ，Vref- for CH0- 輸入
- ❖  $CSCNA=1$ ，啟動輸入掃描功能
- ❖  $CSSL<15:0>=0000\ 0011\ 1111\ 1111$ ，掃描輸入腳為 AN0 ~ AN9
- ❖  $CHOSB<3:0>=N/A$
- ❖  $CHONB=N/A$



# A/D 轉換設定：範例一

- ❖ 自動掃描 AN0 - AN7, 自動轉換, 設成兩組輸出暫存區
  - ◆ 自動掃描 8 類比輸入 - CSCNA = 1, ADCSSL = 0x00FF
  - ◆ 自動轉換 - SSRC<2:0> = 111, ASAM = 1
  - ◆ 設定 SAMC<4:0> 自動取樣時間
  - ◆ 8 次轉換後產生中斷 - SMPI<3:0> = 0111
  - ◆ BUFM = 1 使用兩組資料緩衝器

**Buffer at 1st interrupt**

<b>ADCBUF0</b>	CHO - AN0
<b>ADCBUF1</b>	CHO - AN1
<b>ADCBUF2</b>	CHO - AN2
<b>ADCBUF3</b>	CHO - AN3
<b>ADCBUF4</b>	CHO - AN4
<b>ADCBUF5</b>	CHO - AN5
<b>ADCBUF6</b>	CHO - AN6
<b>ADCBUF7</b>	CHO - AN7

**Buffer at 2nd interrupt**

<b>ADCBUF8</b>	CHO - AN0
<b>ADCBUF9</b>	CHO - AN1
<b>ADCBUFA</b>	CHO - AN2
<b>ADCBUFB</b>	CHO - AN3
<b>ADCBUFC</b>	CHO - AN4
<b>ADCBUFD</b>	CHO - AN5
<b>ADCBUFE</b>	CHO - AN6
<b>ADCBUFF</b>	CHO - AN7

## A/D 轉換設定：範例二

- ❖ 依序掃描 4 類比輸入( AN0,AN1,AN2,AN3)及高速掃描一個類比輸入AN4
  - ◆ 設定 A、B 兩組多工輸入模式：ALTS=1
  - ◆ 多工器 A 將使用四個類比輸入依序掃描 AN0-AN3,   
CSCNA = 1 , ADCSSL = 0x000F
  - ◆ 多工器 B 永遠使用 AN4 為輸入
  - ◆ 轉換八次後產生中斷：SMPI<3:0> = 0111
  - ◆ 連續取樣
  - ◆ 輸入只使用 CH0

ADCBUF0	CH0 - AN0
ADCBUF1	CH0 - AN4
ADCBUF2	CH0 - AN1
ADCBUF3	CH0 - AN4
ADCBUF4	CH0 - AN2
ADCBUF5	CH0 - AN4
ADCBUF6	CH0 - AN3
ADCBUF7	CH0 - AN4



MICROCHIP

# dsPIC30F4011 ADC Hand-On

- ❖ Manual Trigger Mode
- ❖ Auto Trigger Mode
- ❖ Timer3 Trigger Mode
- ❖ External Trigger Mode
- ❖ Motor PWM Trigger Mode

# 各種 AD 轉換觸發模式

利用 **SSRC<2:0>** 來選擇 AD 轉換的觸發信號來源

000 = 手動轉換，SAMP=1 時取樣，清除 SAMP 位元後轉換

111 = 使用內部時序設定取樣時間及轉換時間(自動轉換)

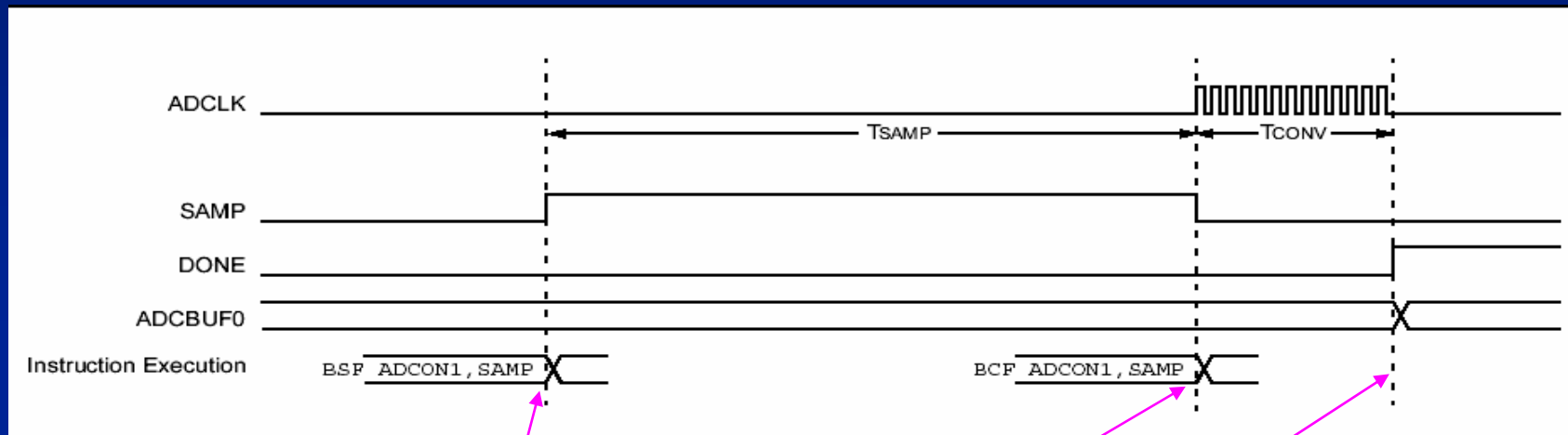
011 = 馬達控制 PWM 間隔結束時，結束取樣ADC開始轉換

010 = Timer 3 計時比較完成後，結束取樣ADC開始轉換

001 = INT0 腳位電位轉態時，結束取樣ADC開始轉換

# AD 轉換 – 手動取樣、手動轉換

使用此模式的基本設定：**SSRC<2:0> = 000**，**ASAM = 0**



```
ADCON1bits.ADON=1;
while(1)
{
```

```
    ADCON1bits.SAMP=1;
```

```
    DelayNuSec(100);
```

```
    ADCON1BITS.SAMP=0;
```

```
    while(!ADCON1bits.DONE);
```

```
    ADCvalue= ADCBUF0;
```

```
}
```

//啓動AD轉換器

//開啓取樣

//延遲100us,取樣電容充電

//取樣結束,開始轉換

//轉換完成了嗎 ?

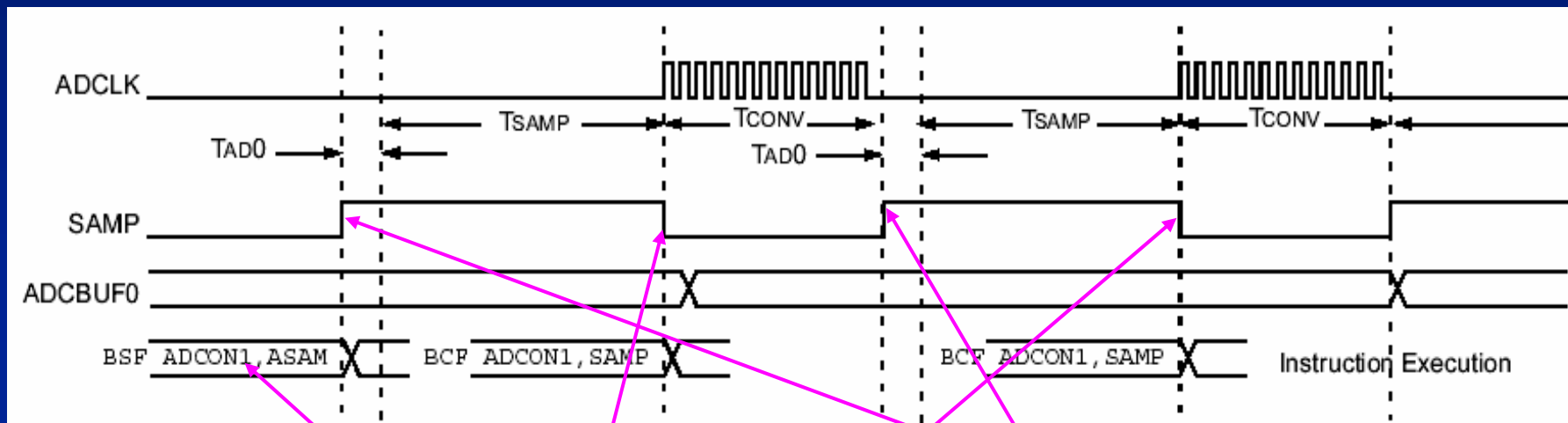
//儲存AD轉換結果



MICROCHIP

# AD 轉換 – 自動取樣，手動轉換

使用此模式的基本設定：**SSRC<2:0> = 000**，**ASAM = 1**



```
ADCON1bits.ADON=1;  
ADCON1bits.ASAM=1;  
while(1)  
{  
    DelayNuSec(100);  
    ADCON1BITS.SAMP=0;  
    while(!ADCON1bits.DONE);  
    ADCvalue= ADCBUF0;  
}
```

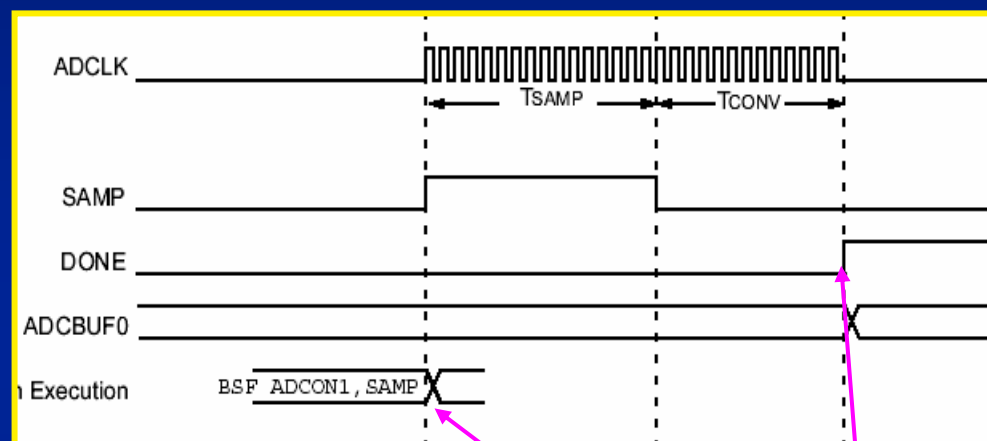
//啓動AD轉換器  
//啓動自動取樣，SAMP位元自動設為 1

//延遲100us，取樣電容充電  
//取樣結束，開始轉換  
//轉換完成了嗎？  
//儲存AD轉換結果，SAMP位元自動設為 1

# AD 轉換 – 手動取樣， $T_{AD}$ 觸發轉換

使用此模式的基本設定：**SSRC<2:0> = 111**，**ASAM = 0**

- ❖ SAMC 位元設定幾個 $T_{AD}$ 的取樣時間 (SAMC=11111 時設需31  $T_{AD}$ )
- ❖ 一但設定 SAMP=1，則會自動取樣再轉換



```
/* Initial the AD Module */
ADPCFG = 0xEFFF; // using AN12
ADCON1 = 0x00E0; // SSRC = 111
ADCHS = 0x000C; // AN12 is CH0 input
ADCSSL = 0x0000; // None Sacn input
ADCON3 = 0x1F02; // 31 Tad , Tad=2Tcy
ADCON2 = 0x0000;
ADCON1bits.ADON=1; // Start AD module
```

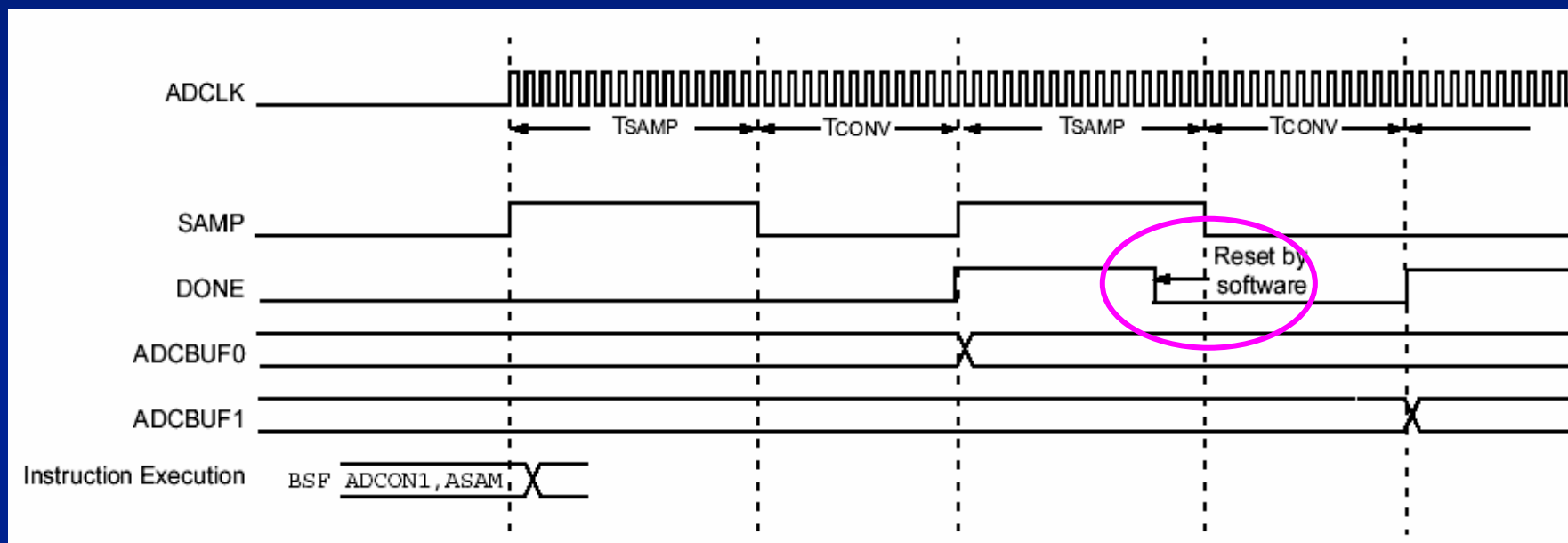
```
While(1)
{
    ADCON1BITS.SAMP=1;
    while(!ADCON1bits.DONE);

    ADCvalue= ADCBUF0;
}
```



# 使用 AD 轉換 – 自行運作模式 (自動取樣與轉換)

使用此模式的基本設定：**SSRC<2:0> = 111**，**ASAM = 1**



- ❖ **SSRC<2:0> = 111** 轉換觸發來自 A/D 的時脈，同時設定 **ASAM=1** (自動取樣與轉換)
- ❖ 在此模式下，**SAMP**會自動設為 1 進行取樣，完成取樣後(**SAMC**設定取樣時間) **SAMP**會自動清為 0 開啓做轉換動作(需14 T<sub>AD</sub>) 週而復始
- ❖ 詳細程式如下頁之說明

# 自動取樣與轉換 – 範例程式

```
#define FILTER_BLOCK_LENGTH 512
int SigIn[FILTER_BLOCK_LENGTH];

InitADC12( );
    ADCON1bits.ADON = 1;

for ( i = 0; i < FILTER_BLOCK_LENGTH; i++ ) // Sample 512 A/D input
{
    while ( !ADCON1bits.DONE );           // wait for end of conversion
    ADCON1bits.DONE = 0;                   // reset DONE bit
    SigIn[i] = ADCBUF0;                     // store sample in SigIn buffer
}
```



**MICROCHIP**

```
void InitADC12(void)
{
    IFS0bits.ADIF = 0;           // clear A/D interrupt flag
    IEC0bits.ADIE = 0;          // disable A/D interrupt
    ADCON1bits.ADON = 0;        // turn off the A/D converter

    ADPCFG = 0xFFF7; // PCFG3 = 0, only AN3 (RB3) input pin in analog mode
    ADCHS = 0x0003; // CH0NB = 0, CH0SB = 0000, CH0SB = Vref-, CH0SA = AN3
    ADCON1 = 0x03E4; // ADON=0, FORM=Frational, SSRC=AD clock Trigger
                    // ASAM=Auto Sampling & Conversion
    ADCON2 = 0x0000; // VCFG = AVdd and AVss are used, CSCNA = disable scanning
                    // SMPI = 0000 , SMPI = 16-word buffer , ALTS = use MUX A

    ADCON3 = 0x1F3F; // 0001 1111 0011 1111
    /******
    /* SAMC = 11111, sampling time is 31*Tad */
    /* ADRC = 0, A/D clock derived from system clock */
    /* ADCS = 111111, Tad = (ADCS<5:0>+1)* Tcy/2, Tconv=14Tad */
    /*-----*/
    /* Tad= 64*33.91nS/2 = 1.08512uS , Tconv= 15.1968uS */
    /* A/D = Tsample+Tconv= (32+14)1.08512= 49.915uS = 50uS(20KHz) */
    /******

    ADCSSL = 0x0000; // 0000 0000 0000 0000
}
```

## 使用 AD 轉換器 – Timer 3 觸發

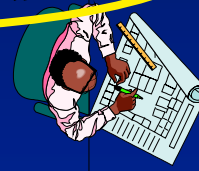
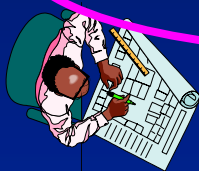
使用此模式的基本設定： $SSRC<2:0> = 010$ ， $ASAM = 1$ ， $SAMP=1$

### ❖ 使用 Timer 3 計時器觸發中斷

- ◆ 使用Timer3中斷觸發可以使用在轉換速度需求較慢的應用,例如:音頻的取樣 (8K 的取樣時間)
  - ◆ 需較快的自動轉換可使用內部時脈驅動方式 ( $SSRC<2:0>=111$ )
- ◆ 每次的 Timer 3 觸發會自動清除  $SAMP=0$  做 AD 的自動轉換，轉換完成後又會因  $ASAM = 1$  而設定  $SAMP=1$  做自動取樣，待下次的 Timer 3 觸發時再度清為零做 AD 轉換，以上動作會週而復始的進行
- ◆ 底下的範例是藉由 Timer 3 的下降緣發使 AD 的產生中斷 (Timer 3 並未設定中斷致能)



MICROCHIP



# A/D 轉換器

## 自動取樣 & Timer3 轉換

類比輸入

取樣時間

轉換時間

取樣時間

轉換時間

Timer3 計時比較  
完成後自動清除  
SAMP開始轉換

上一次轉換完成後  
硬體自動取樣  
(SAMP 被設為 1)  
因 ASAM=1

Timer3 計時比較  
完成後自動清除  
SAMP開始轉換

ADCON1

ADON=1	-	ADSIDL	-	-	-	FORM<1:0>
--------	---	--------	---	---	---	-----------

SSRC=010	-	SIMSAM	ASAM=1	SAMP	DONE
----------	---	--------	--------	------	------

Bit 0

# Timer 3 觸發 – 範例程式

```
void _ISR _ADCInterrupt(void)
{
    IFS0bits.ADIF = 0 ;    // 清除 ADIF 中斷旗號
    ADC_Buf = ADCBUF0 ;    // 讀取 AD 的轉換值
    LED16 = !LED16 ;       // LED16 轉態一次供量取訊號除錯用
}
```

本範例是 **Timer 3** 觸發 **AD** 轉換，待 **AD** 轉換完成時立即產生中斷  
注意！**SAMP** 位元硬體會自動設定(取樣)與清除(轉換)，軟體毋需  
設定此位元



MICROCHIP

## Timer 3 & ADC 的初始設定

```
Void Timer3_Initial ( void )
{
    ConfigIntTimer3 ( T3_INT_PRIOR_7 & T3_INT_OFF );      //中斷禁能
    OpenTimer3 ( T3_ON & T3_IDLE_STOP & T3_GATE_OFF & T3_PS_1_1 &
                T3_SOURCE_INT , (((long)FCY/1000)) );      //計時=1mS, 啟動 Timer3
}

void ADC10_Initial (void)
{
    ADPCFG = 0xFF7F;      // AN7/RB7 為類比電壓輸入腳，其它為一般 I/O
    ADCON1 = 0x0046;      // 0b0000 0000 0100 0110
                          // 設定 Timer3 為 AD 轉換的觸發來源
                          // A/D 採用自動取樣自動轉換模式
    ADCON2 = 0x0000;      // 參考電壓：Vref+ = Vdd, Vref- = Vss
                          //不採用輸入掃描方式，SMPI=000（每次轉換完成就產生中斷）
    ADCSSL = 0x0000;      // 不採用輸入掃描方式
    ADCON3 = 0x1F3F;      // TAD = 8 Tcy , SAMC = 15 TAD
    ADCHS = 0x0007;      // CH0正端輸入選擇AN7，負端輸入為Vss
    IEC0bits.ADIE = 1;    // 打開AD的中斷
    IPC2bits.ADIP = 7;    // 中斷等級=7（最高優先權中斷等級）
    ADCON1bits.ADON = 1;  // 啟動 AD
}
```





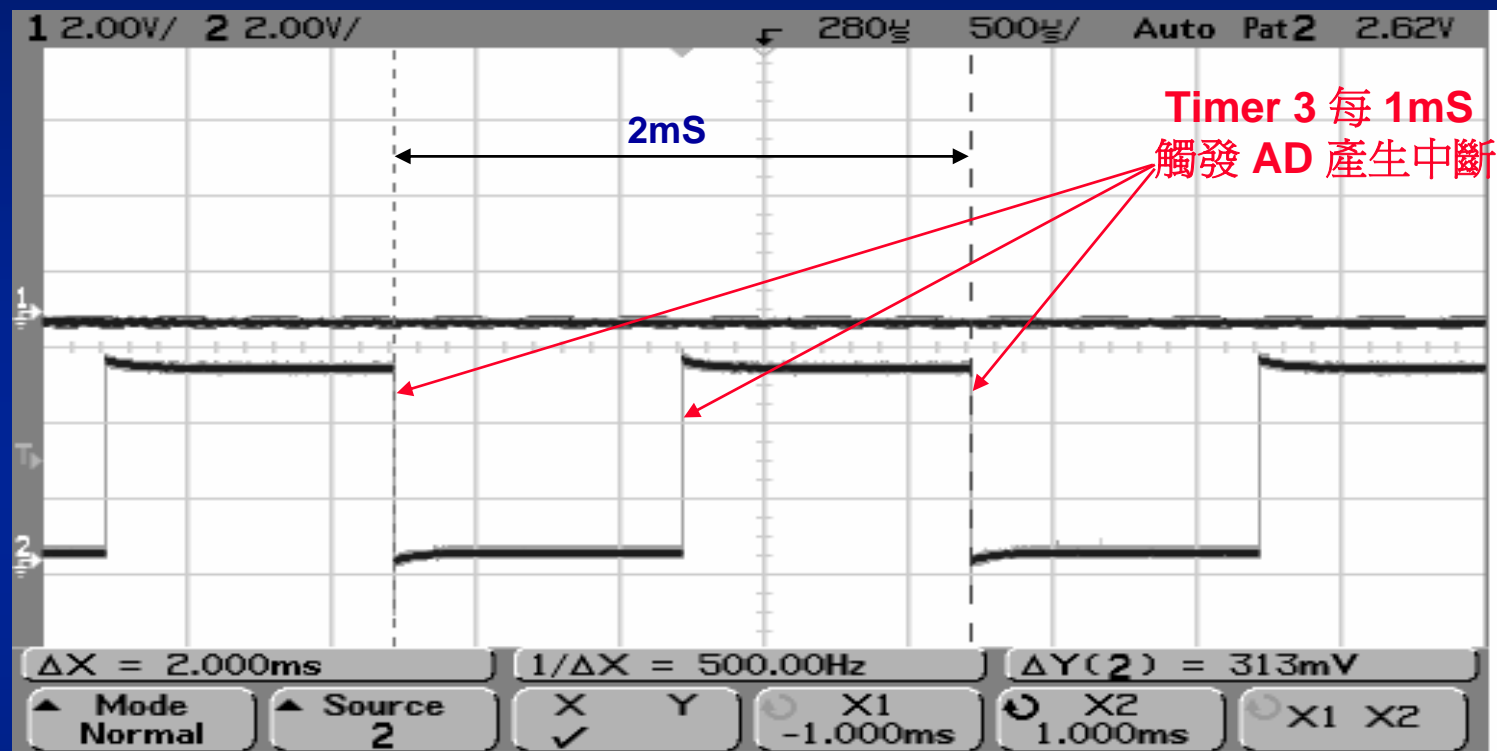
# 練習一 使用 Timer3 來觸發 ADC 的轉換

- ❖ 練習一的動作 – 讀取 VR1 的位準
  - ◆ 將 Timer 3 規劃成週期為 1ms
  - ◆ 將 ADC 規劃成使用 Timer 3 觸發的工作模式
  - ◆ 每次 ADC Interrupt 時
    - ◆ 將 ADCBUF0 讀進 MyADC
    - ◆ 設定旗標 Flags.ADC\_DONE
    - ◆ 將 APP020 上的 LED1 反向一次
  - ◆ 主程式看到 ADC\_DONE 旗標為一時
    - ◆ 將 MyADC 的值顯示於 LCD
    - ◆ 清除 ADC\_DONE



# Timer 3 觸發波形量測

LED16



Timer-3 Period = 1ms

RFO 於每次 ADC interrupt 時 Toggle  
於是波器可看到 RFO 每 1ms 轉態

# 使用 AD 轉換器 – 外部 INTO 觸發

使用此模式的基本設定：**SSRC<2:0> = 001**，**ASAM = 1**，**SAMP=1**

## ❖ 使用 INTO 外部觸發中斷

- ◆ INTO 可設定上昇緣或下降緣來觸發 AD 的轉換
  - ◆ **INTCON2<INTOEP>位元=1**，設為下降緣
  - ◆ **INTCON2<INTOEP>位元=0**，設為上昇緣
- ◆ 每次的 INTO 觸發會自動清除 **SAMP=0** 做 AD 的自動轉換，轉換完成後又會設定 **SAMP=1** 做自動取樣，待下次的 INTO 觸發時再度清為零做 AD 轉換，以上動作會週而復始的進行
- ◆ 底下的範例是藉由 INTO 的下降緣發使 AD 的產生中斷（INTO 並未設定中斷致能）

# 外部 INTO 觸發 – 範例程式

```
void _ISR _ADCInterrupt(void)
{
    IFS0bits.ADIF = 0 ;    // 清除 ADIF 中斷旗號
    ADC_Buf = ADCBUF0 ;    // 讀取 AD 的轉換值
    LED16 = !LED16 ;       // LED16 轉態一次供量取訊號除錯用
}
```

本範例是 **INT0** 觸發 **AD** 轉換，待 **AD** 轉換完成時立即產生中斷  
注意！**SAMP** 位元硬體會自動設定(取樣)與清除(轉換)，軟體毋需  
設定此位元

# INT0 & ADC 的初始設定

```

void INTO_Initial( void )
{
    TRISEbits.TRISE8 = 1 ;    // Set INT0 for a input
    INTCON2bits.INT0EP = 1 ;  // INT0 interrupt on negative edge
    IECObits.INT0IE = 0 ;
}

void ADC10_Initial(void)
{
    ADPCFG = 0xFF7F;          // AN7/RB7 為類比電壓輸入腳，其它為一般 I/O
    ADCON1 = 0x0026;          // 0b0000 0000 0010 0110
                                // 設定 INT0 為 AD 轉換的觸發來源
                                // A/D 採用自動取樣自動轉換模式
    ADCON2 = 0x0000;          // 參考電壓：Vref+ = Vdd, Vref- = Vss
                                // 不採用輸入掃描方式， SMPI = 000 ( 每次轉換完成就產生中斷 )
    ADCSSL = 0x0000;          // 不採用輸入掃描方式
    ADCON3 = 0x1F3F;          // TAD = 8 Tcy , SAMC = 15 TAD
    ADCHS = 0x0007 ;          // CHO正端輸入選擇AN7，負端輸入為Vss
    IECObits.ADIE = 1 ;       // 打開AD的中斷
    IPC2bits.ADIP = 7 ;       // 中斷等級=7 ( 最高優先權中斷等級 )
    ADCON1bits.ADON = 1;      // 啟動 AD
}
    
```



## 練習二 使用 INTO 來觸發 ADC 的轉換

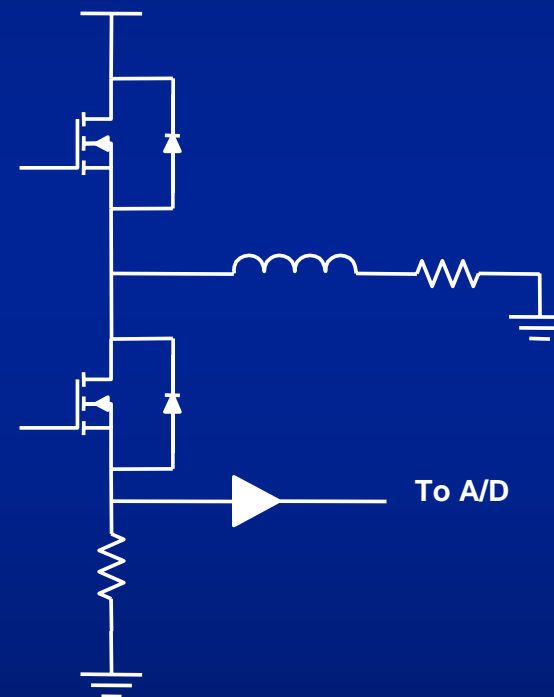
- ❖ 練習二的動作 – 轉換 VR1 的輸入位準
  - ◆ 將 INTO/FLTA ( 按鍵 SW5 ) 規劃成下降緣觸發
    - ◆ 不要 Enable 其中斷
  - ◆ 將 ADC 規劃成使用 INTO 觸發的工作模式
  - ◆ 每次 ADC Interrupt 時
    - ◆ 將 ADCBUF0 讀進 MyADC
    - ◆ 設定旗標 Flags.INT\_ACTIVE
    - ◆ 將 APP020 上的 LED1 反向一次
  - ◆ 主程式看到 INT\_ACTIVE 旗標為 “1” 時：
    - ◆ 將 MyADC 的值顯示於 LCD
    - ◆ 清除 INT\_ACTIVE

# 使用 AD 轉換器 – PWM 觸發

使用此模式的基本設定： $SSRC\langle 2:0 \rangle = 011$ ， $ASAM = 1$ ， $SAMP=1$

## ❖ 使用 Motor PWM 觸發中斷

- ◆ SEVTCMP 暫存器設定 A/D 觸發轉換的時間值
- ◆ SEVTDIR $\langle 15 \rangle$  設定在 PTMR 暫存器往上數或往下數時用 SEVTCMP 的內容值來觸發 ADC
- ◆ 如此可確保 A/D 可正確地偵測 PWM Low 導通時的電流



# Motor PWM 觸發 – 範例程式

```
void _ISR _ADCInterrupt(void)
{
    IFS0bits.ADIF = 0 ;    // 清除 ADIF 中斷旗號
    ADC_Buf = ADCBUF0 ;    // 讀取 AD 的轉換值
    LED16 = !LED16 ;       // LED16 轉態一次供量取訊號除錯用
}
```

本範例是 **Motor PWM 觸發 AD 轉換**，待 **AD 轉換完成時立即產生中斷**  
注意！**SAMP** 位元硬體會自動設定(取樣)與清除(轉換)，軟體毋需設定此位元



MICROCHIP

## 10-bit ADC 的初始設定

```
void ADC10_Initial(void)
{
```

```
    ADPCFG = 0xFF7F;    // AN7/RB7 為類比電壓輸入腳，其它為一般 I/O
    ADCON1 = 0x0066;    // 0b0000 0000 0110 0110
                        // 設定 Motor PWM 為 AD 轉換的觸發來源
                        // A/D 採用自動取樣自動轉換模式
    ADCON2 = 0x0000;    // 參考電壓：Vref+ = Vdd, Vref- = Vss
                        // 不採用輸入掃描方式，SMPI=000（每次轉換完成就產生中斷）
    ADCSSL = 0x0000;    // 不採用輸入掃描方式
    ADCON3 = 0x1F3F;    // TAD = 8 Tcy, SAMC = 15 TAD
    ADCHS = 0x07;       // CH0正端輸入選擇AN7，負端輸入為Vss

    IEC0bits.ADIE = 1;  // 打開AD的中斷

    IPC2bits.ADIP = 7;  // 中斷等級=7（最高優先權中斷等級）

    ADCON1bits.ADON = 1; // 啟動 AD
}
```



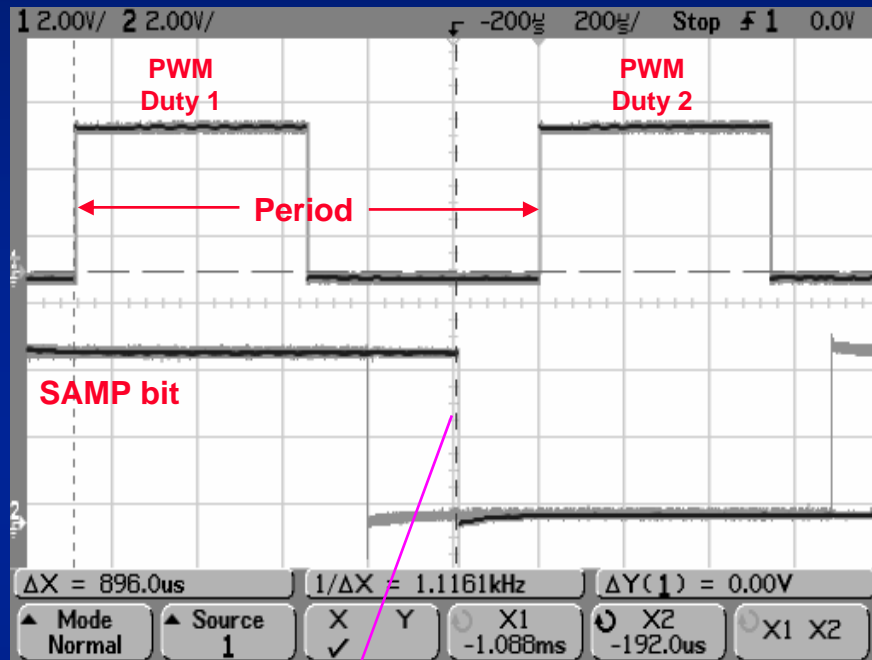


**MICROCHIP**

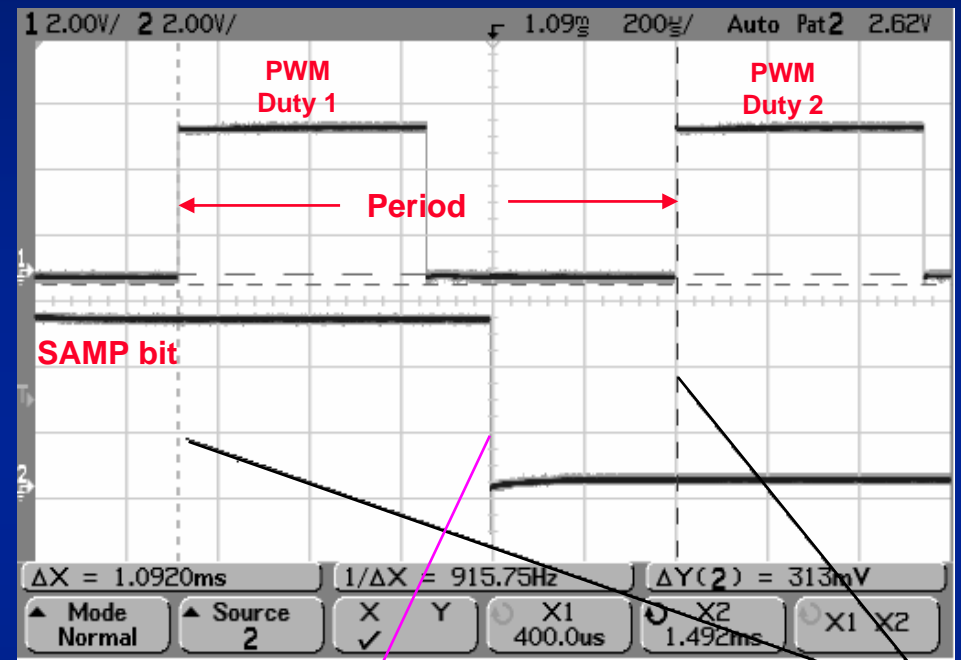
## PWM1 的初始設定

```
void    MotPWM_Initial(void)
{
    IEC2bits.PWMIE = 0 ;           // Disable PWM Interrupt !!
    IEC2bits.FLTAE = 0 ;
    OVDCON = 0xff00 ;             // Active all PWM OUTPUT !!
    TRISE = 0xffc0 ;
    PTCON = 0xa008 ;              // Configure as 0b1010 0000 0000 1000
                                   // PWM Time Base Prescale = 1:16
                                   // PWM Time Base OP in free running Mode
    PWMCON1 = 0x0077 ;            // Configure as 0b00000000001110111
                                   // PWM I/O in complementary Mode and only PWM1L/H
                                   // as PWM output
    PWMCON2 = 0x0000 ;            // Configure as 0b0000000000000000
    DTCON1 = 0x0101 ;             // Configure as 0b0000 0010 0000 0010 ;
    FLTACON = 0x0000 ;
    IPC9bits.PWMIP = 6 ;
    PTPER = 1000 ;                // PWM Time Base Period Register
    PDC1 = 1000 ;
    PDC2 = 1000 ;
    PDC3 = 1000 ;
}
```

# Motor PWM 觸發波形量測



PWM Period = 1.092mS  
PTPTR = 1000  
SEVTCMP = 800



PWM Period = 1.092mS  
PTPTR = 1000  
SEVTCMP = 600

1.092mS