Ch7-ADC MODULE

BỘ CHUYỂN ĐỔI ADC
(Analog-to-Digital)

Kiến thức sinh viên cần đạt được:

Năm được các tính năng, cấu trúc và nguyên lý hoạt động của modul ADC, các thanh ghi điều khiển liên quan

Giải thích được khái niệm và chức năng của điện áp tham chiếu.

Thiết lập được điện áp tham chiếu trong và ngoài cho khối ADC của vi điều khiển.

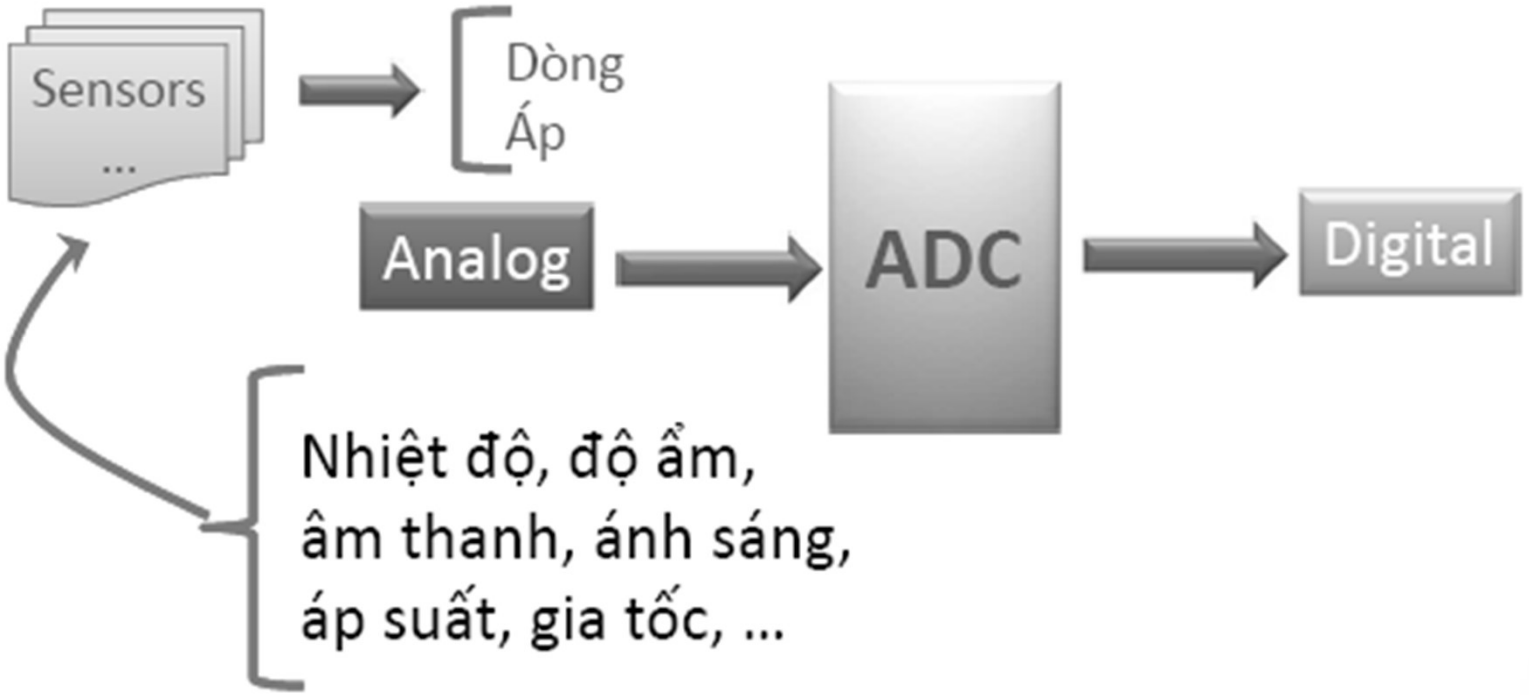
Liệt kê được các bước thiết lập đo ADC cho một hoặc nhiều kênh.

Thiết lập được công thức tính ADC 8-bit , và ADC 10-bit ở chế độ định dạng canh trái và canh phải .

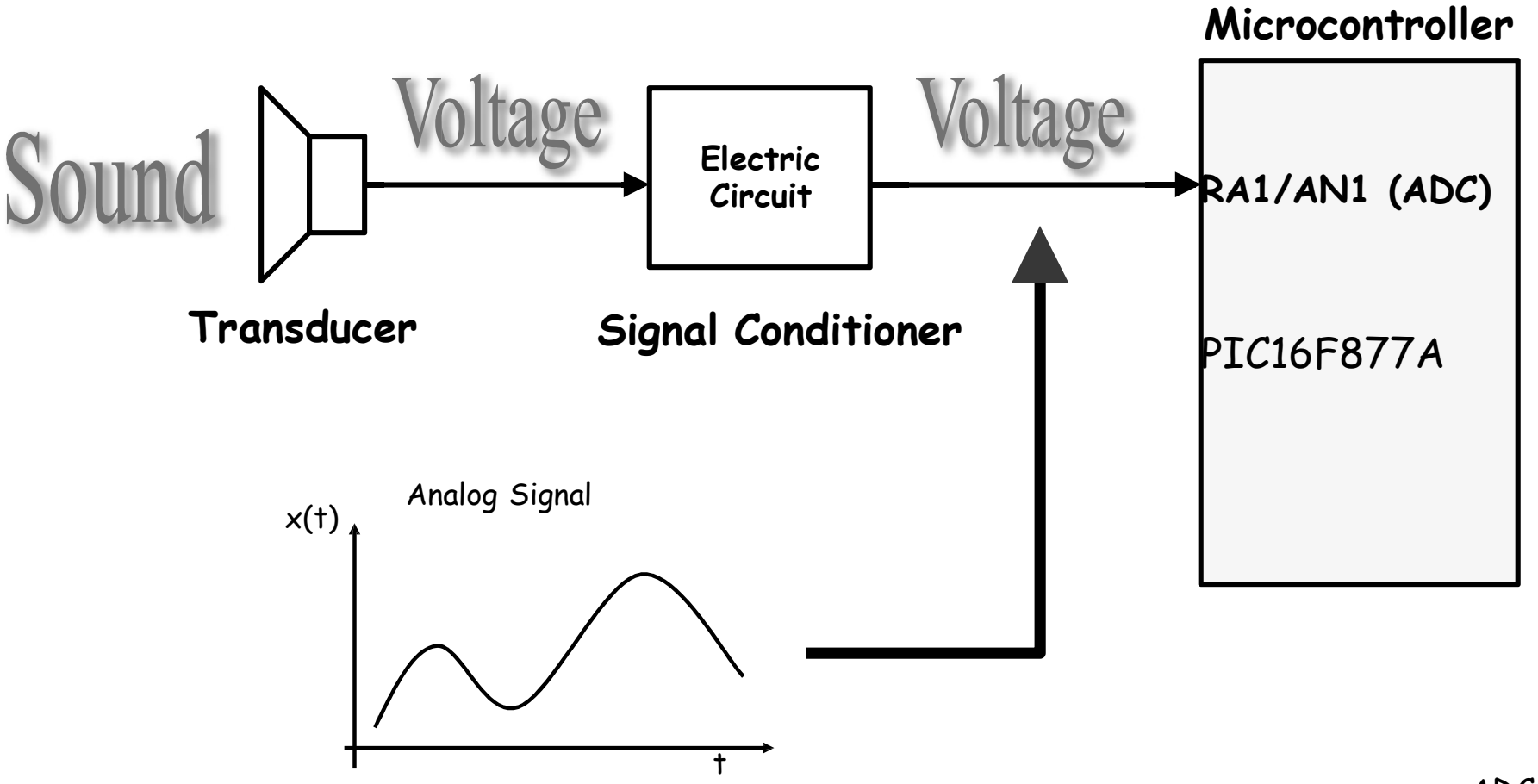
Tín toán được giá trị tín hiệu tương tự thu được thông qua giá trị của thanh ghi ADRESL, ADRESH.

Giới thiệu ADC

- **Khái niệm:** ADC - bộ chuyển đổi tín hiệu tương tự thành tín hiệu số



Analog-to-Digital Conversion (ADC)



What is an ADC?

An ADC (Analog-to-Digital-Converter) is a circuit which gets an **analog voltage signal** and provides (to software) a **variable** proportional to the input signal.

An ADC is characterised by:

-The **range** (in Volts) of the input signal (typical [0, 5V] or [0, 3.3V]).

-The **resolution** (in **bits**) of the converter.

Example:

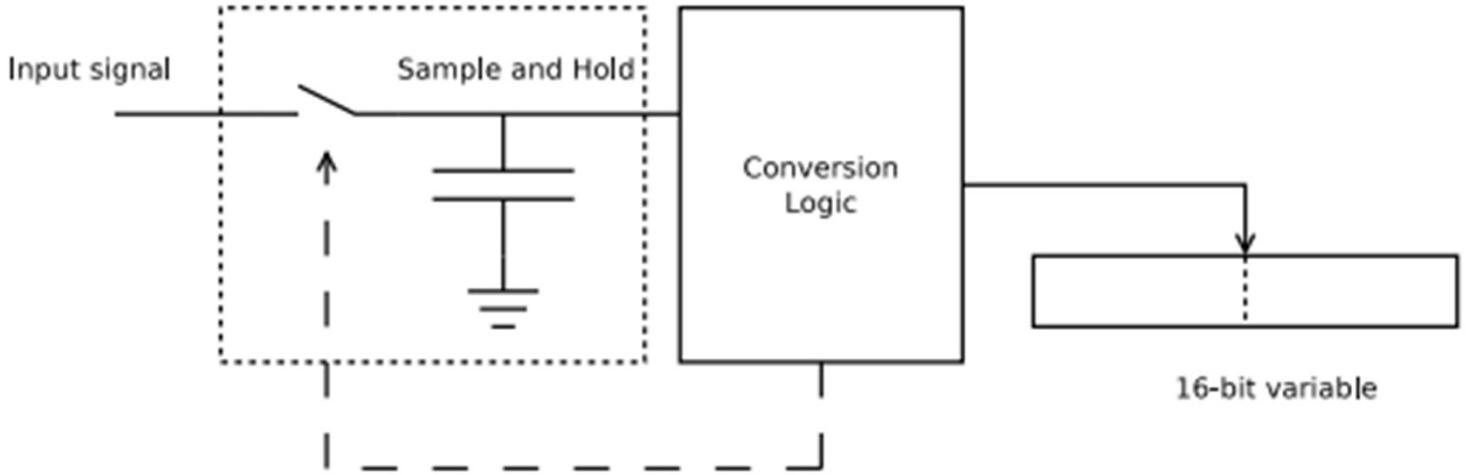
-**Range** = [0, 5V]

-**Resolution** = 12 bits

results in range [0, 212 - 1] = [0, 4095]

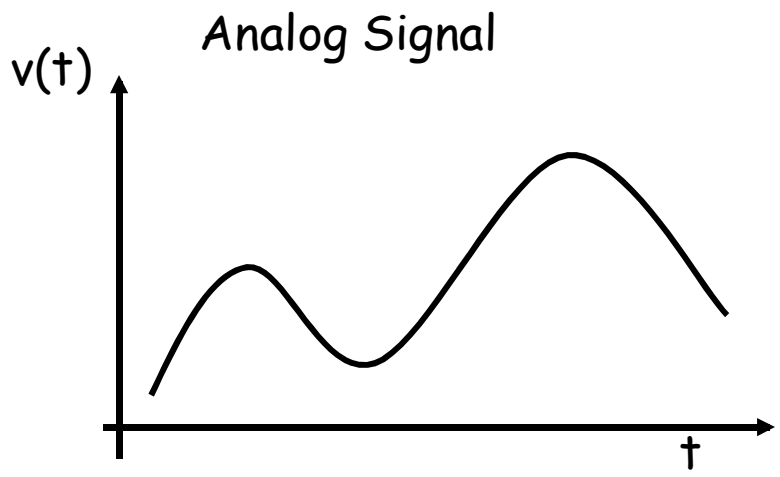
-0V $\rightarrow$ 0 2.5V $\rightarrow$ 2048 5V $\rightarrow$ 4095

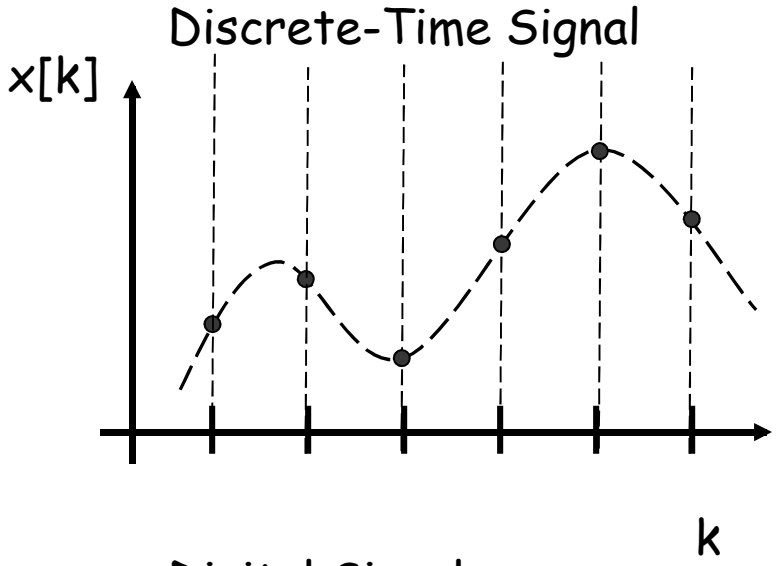
ADC: Basic working scheme

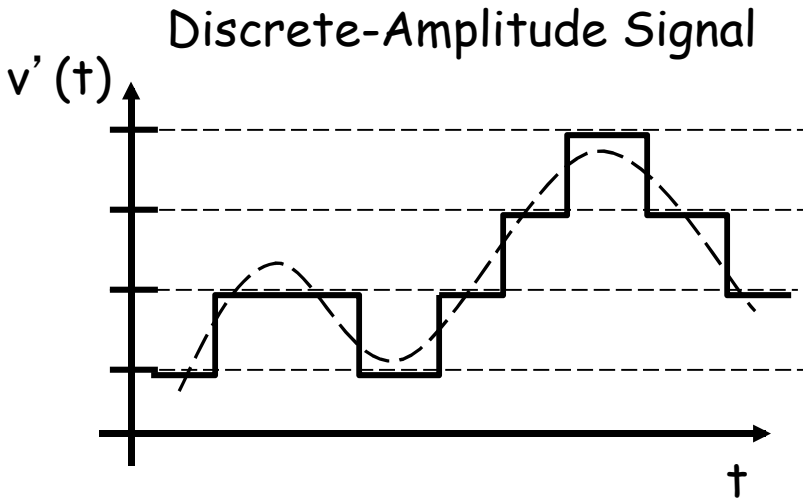


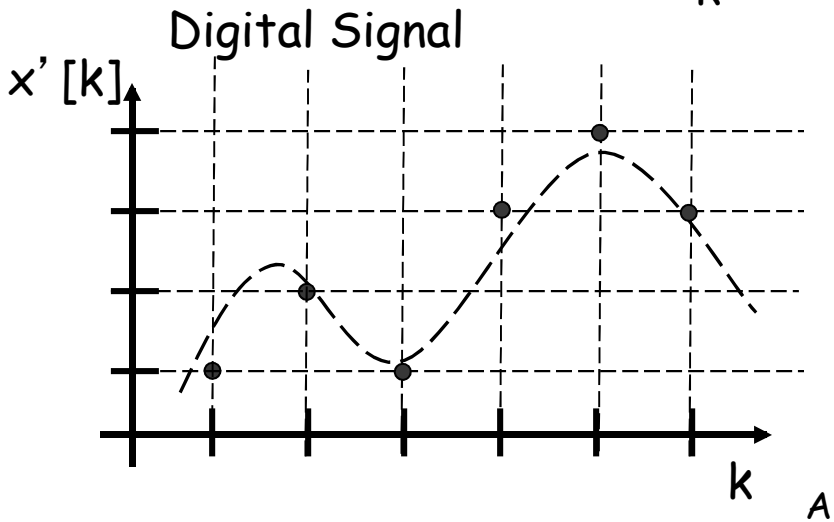
- **1 Sample**: the signal is *sampled* by closing the switch and charging the capacitor.
  2 Conversion: the switch is open and the sampled signal is *converted*; the result is stored in the 16-bit variable.
  3 End-of-conversion: a proper bit is set to signal that the operation has been done.

Analog-to-Digital Conversion



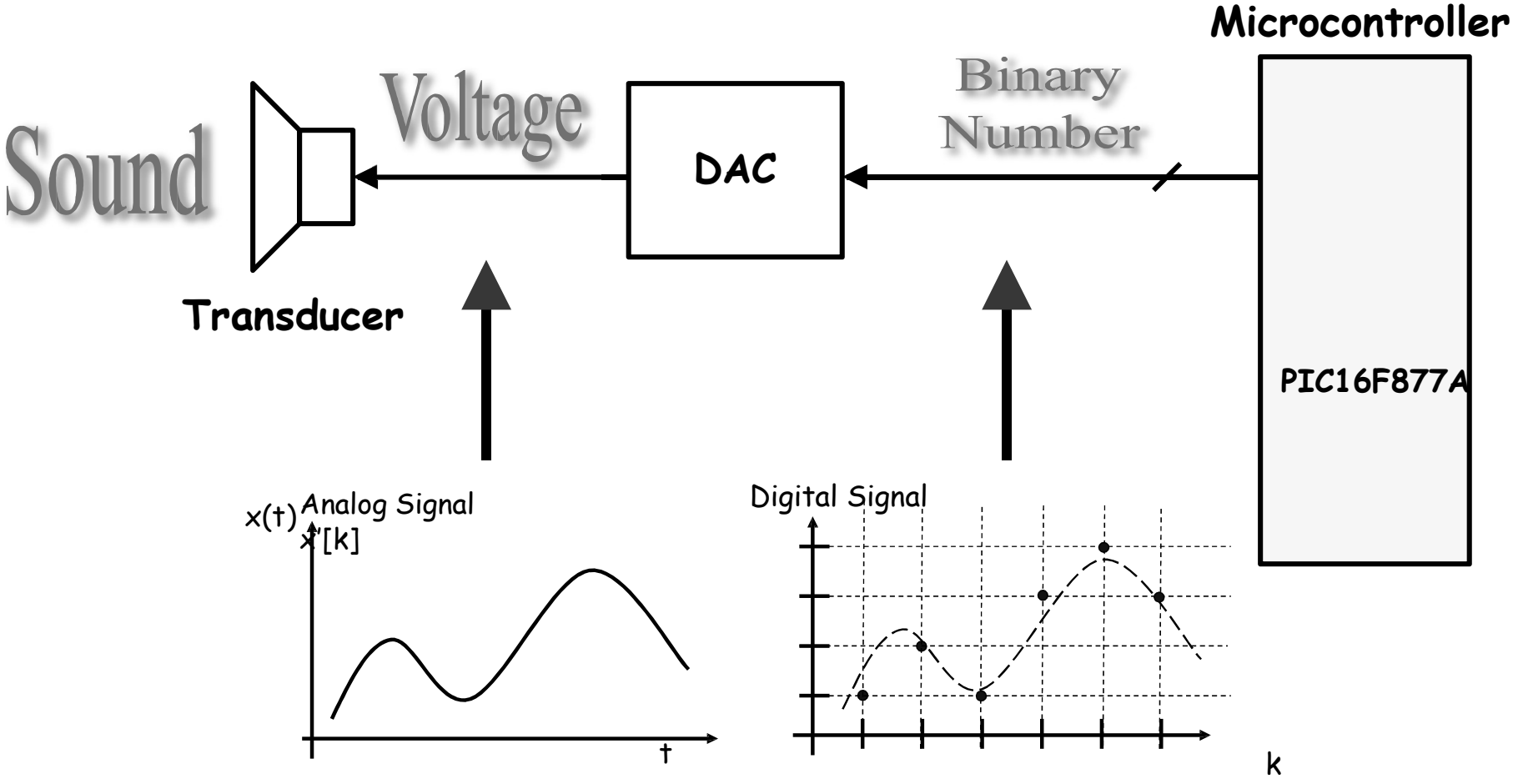






Digital-to-Analog Conversion (ADC)

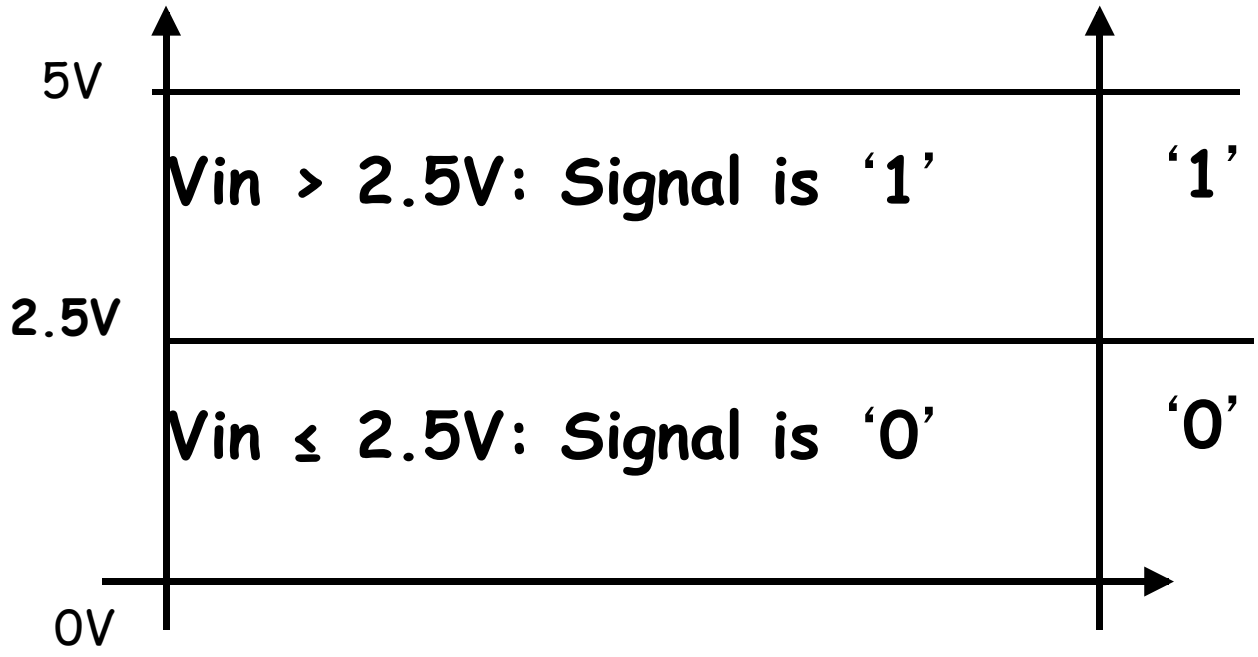
Example



PIC Microcontroller does NOT have a built-in DAC!!!

Examples of ADC

1-bit Example

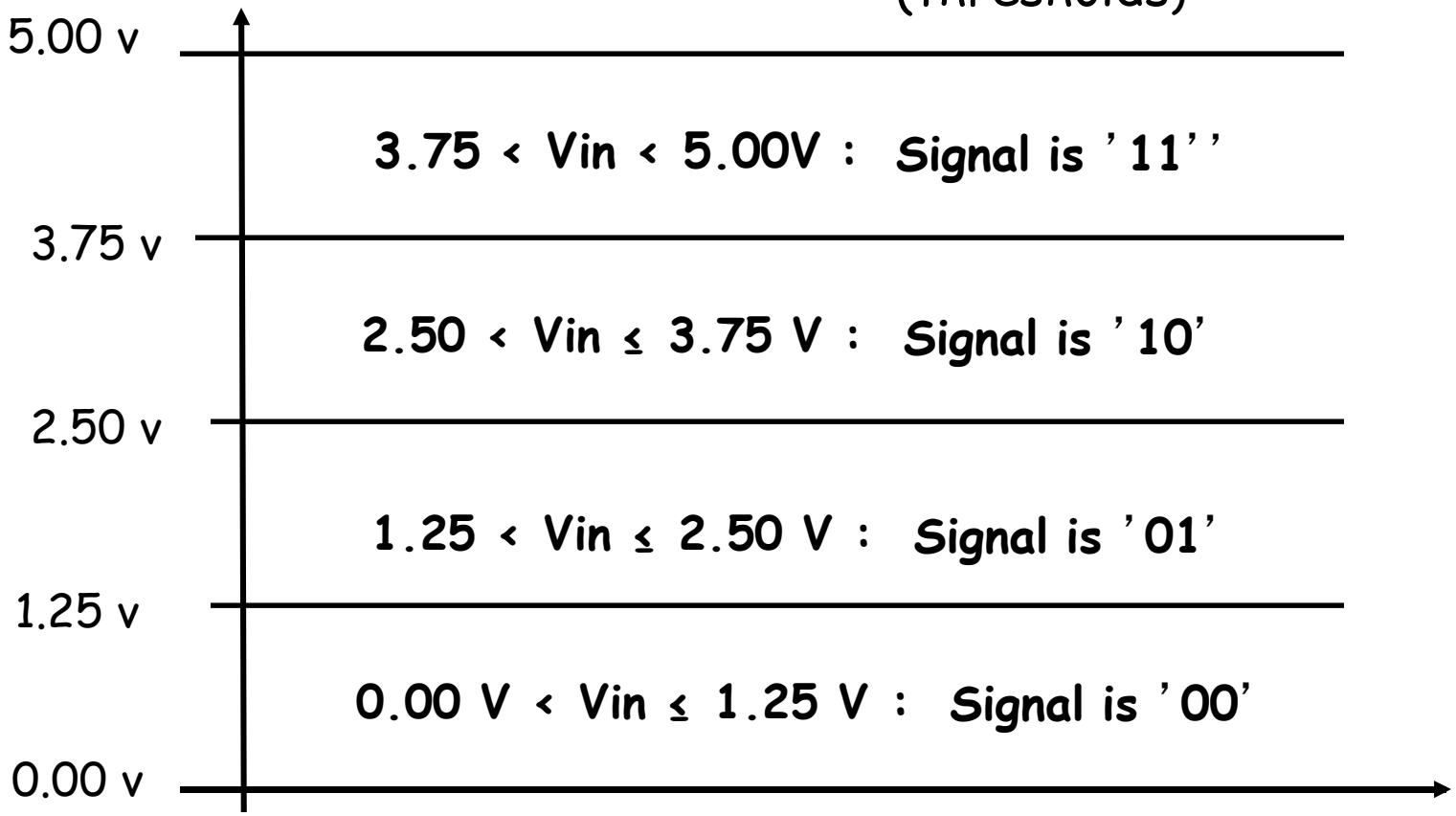


1-bit => 2^1 = 2 levels => 1 threshold

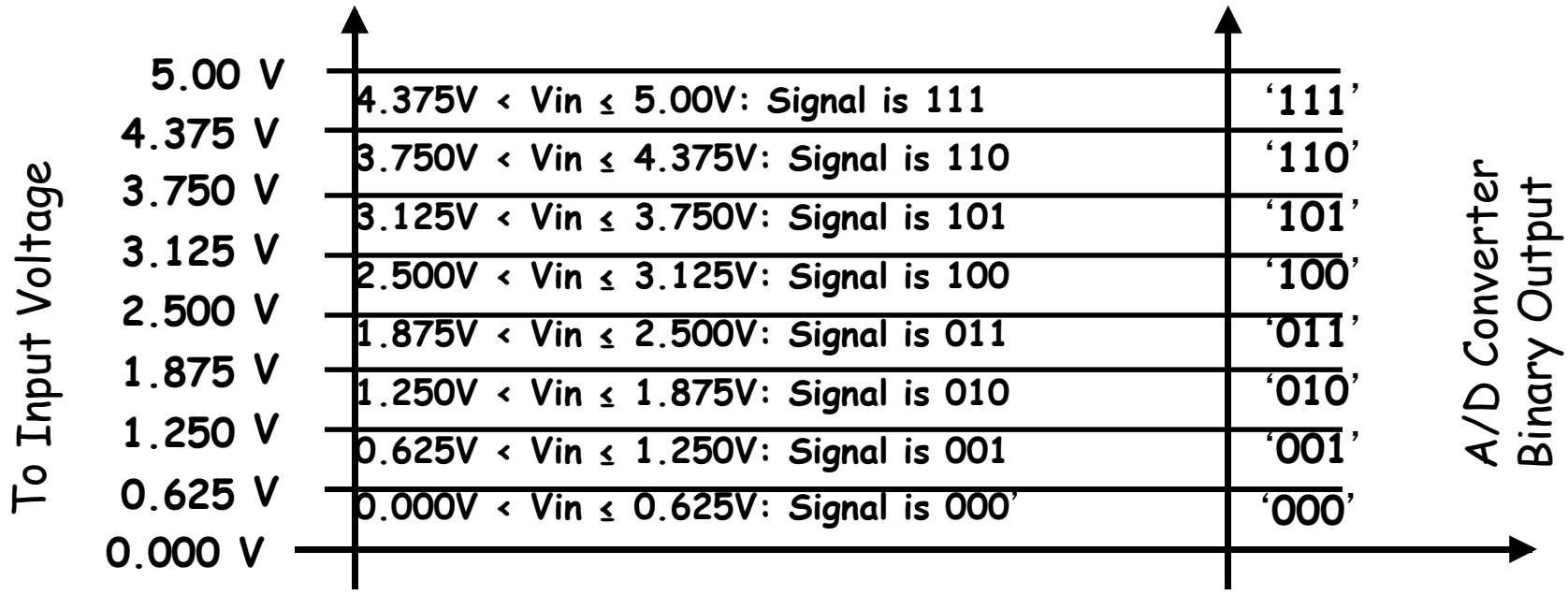
ADC
2-bit Example

2-bit => 2^2 = 4 levels => 3 thresholds

5 v / 4 levels = 1.25 each increment (thresholds)



ADC : 3-bit Example



3-bits => 2^3 = 8 levels => 7 thresholds (5v/8 = 0.625)

ADC The PIC16F877A Microcontroller

- 10-bits:
  - 2^{10} = 1024 levels => 1023 thresholds
  - Resolution = 5V/1024 = 0.0048828125 V
  - Thresholds:
    - 0.0048828125 V
    - 0.009765625 V
    - 0.0146484375 V
    - Etc.

BM DTMT CNĐT

The ADC Circuit of PIC16F877A

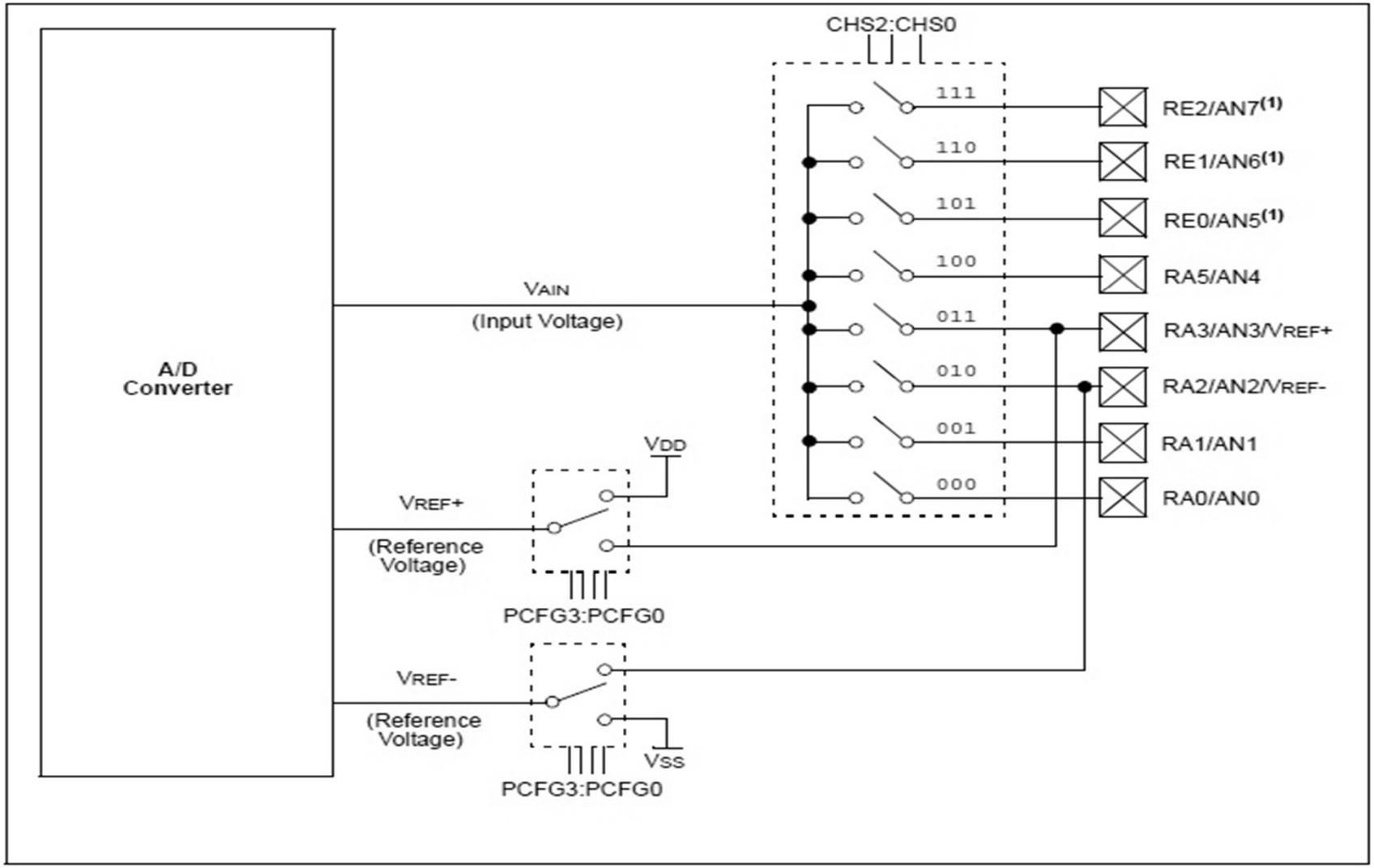
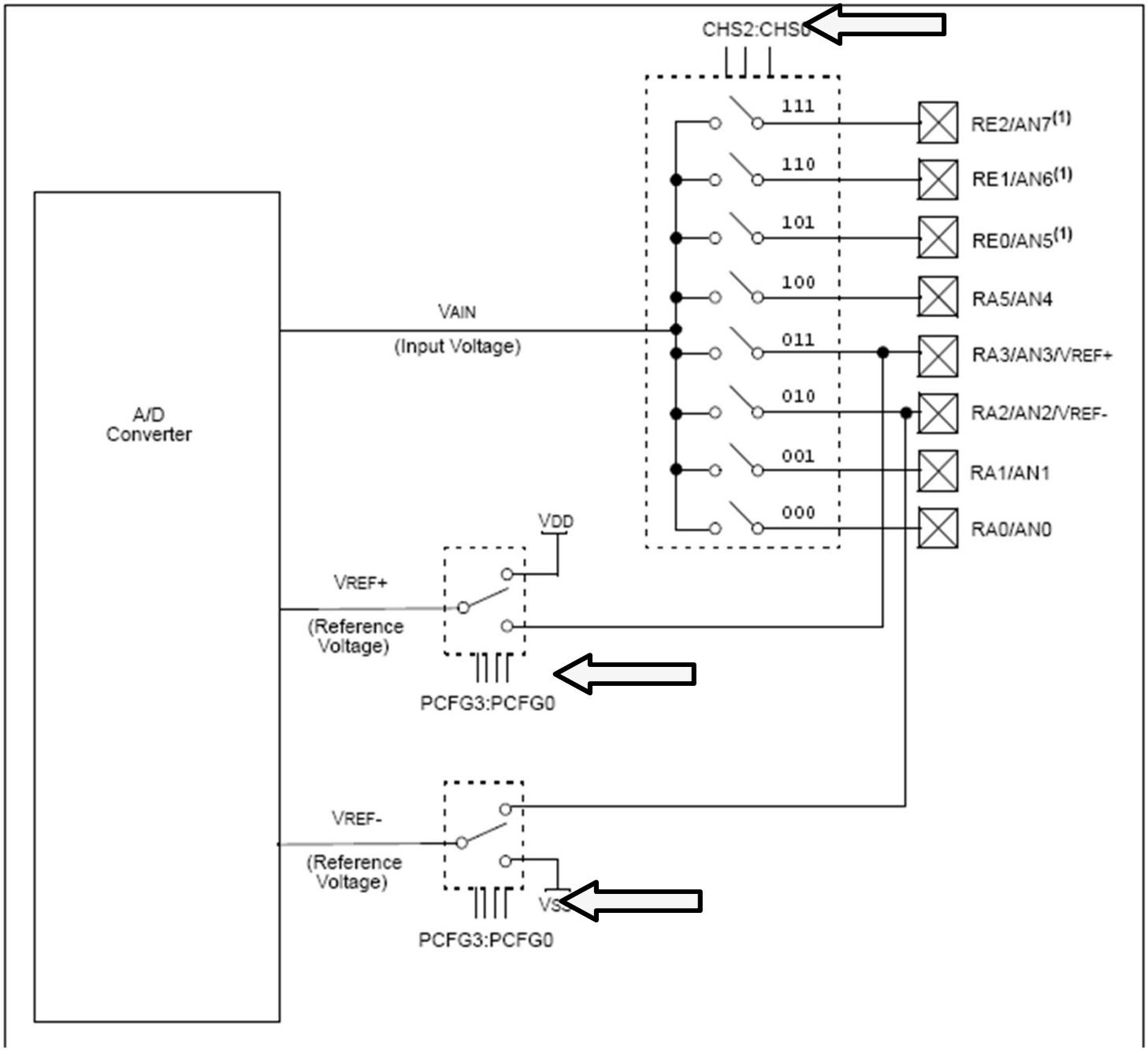
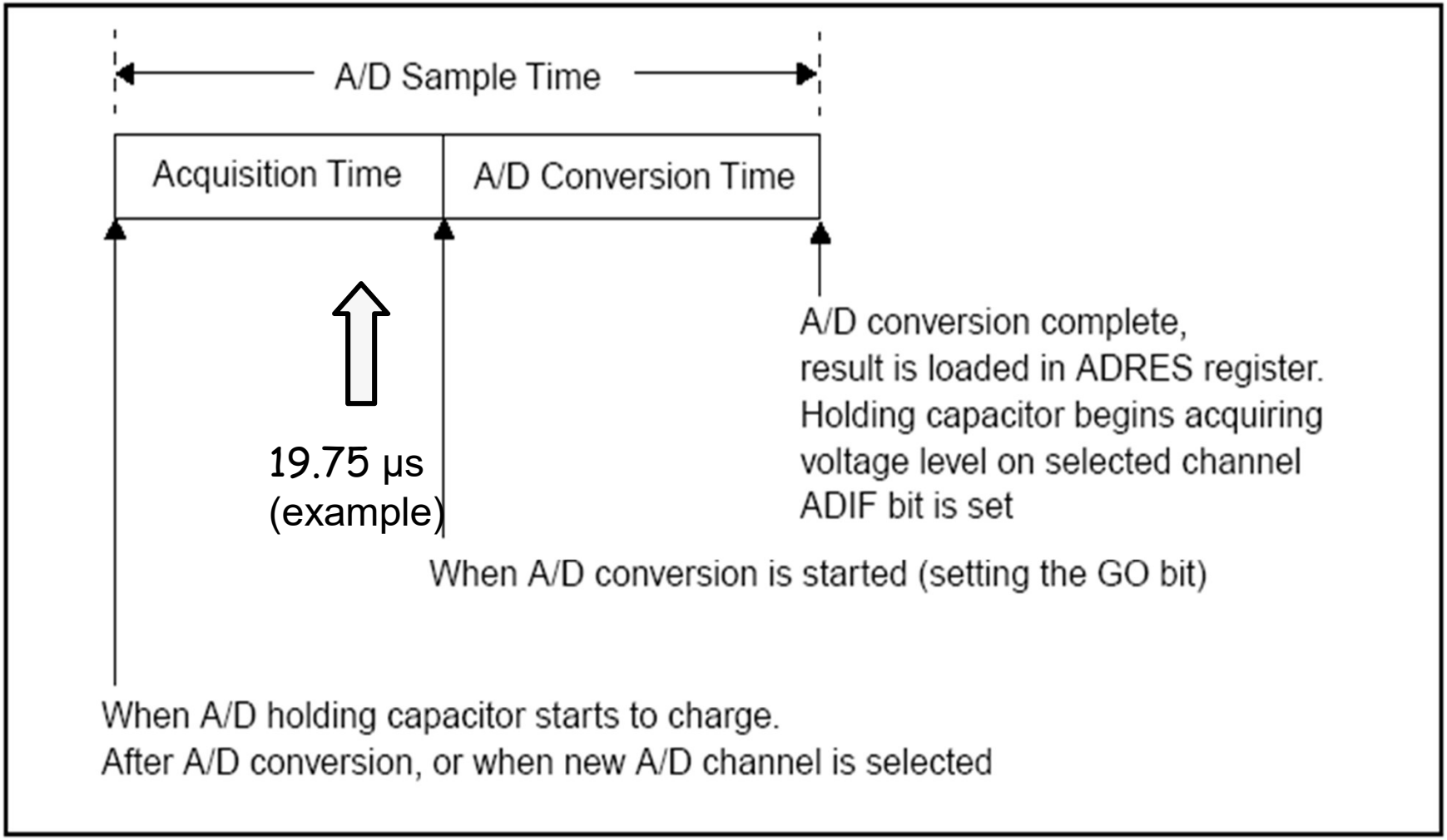


FIGURE 11-1: A/D BLOCK DIAGRAM



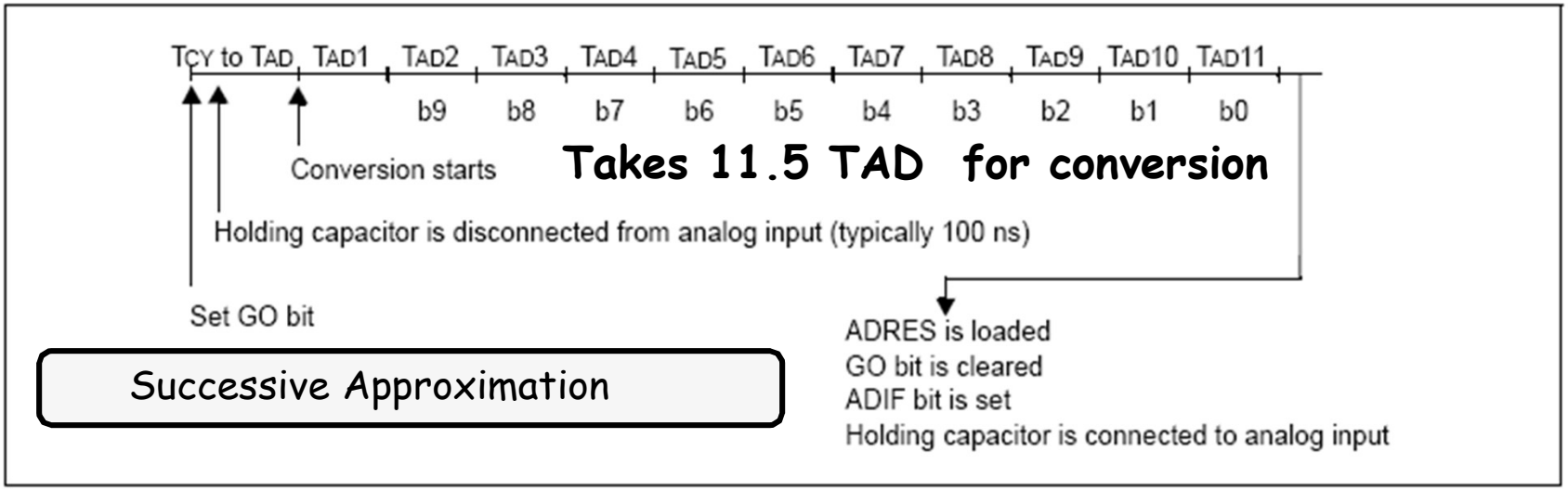
ADC Timing

Figure 23-2: A/D Conversion Sequence



ADC Conversion Time for Successive Approx. Converter

FIGURE 11-3: A/D CONVERSION TAD CYCLES



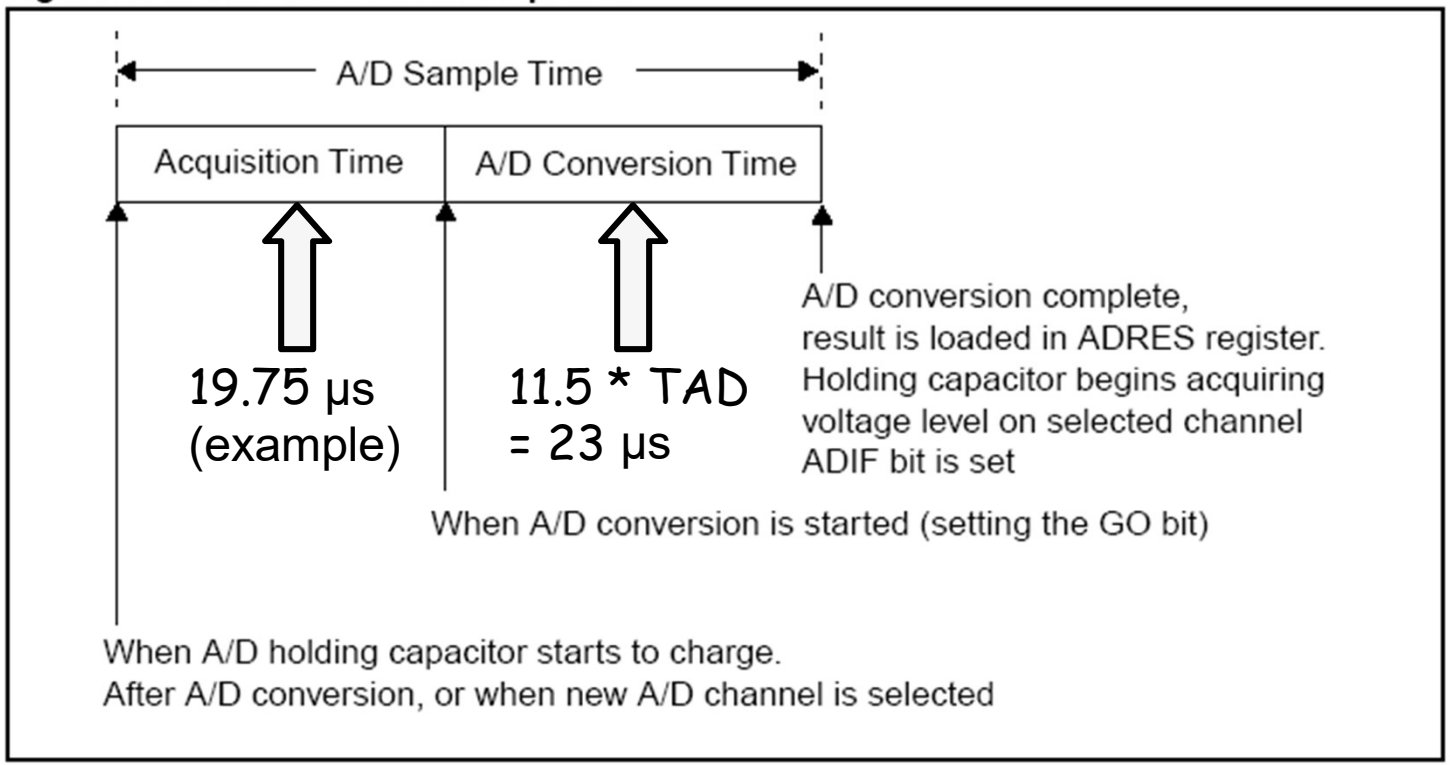
T_{AD} must be ≥ 1.6μs

TABLE 11-1: TAD vs. MAXIMUM DEVICE OPERATING FREQUENCIES (STANDARD DEVICES (C))

AD Clock Source (TAD)		Maximum Device Frequency
Operation	ADCS1:ADCS0	Max.
2Tosc	00	1.25 MHz
8Tosc	01	5 MHz
32Tosc	10	20 MHz
RC(1, 2, 3)	11	(Note 1)

Time for A/D Conversion - Summary

Figure 23-2: A/D Conversion Sequence



Note – You must wait at least 2 * TAD before next acquisition is started.

Các thanh ghi liên quan

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on MCLR, WDT
0Bh,8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
1Eh	ADRESH	A/D Result Register High Byte								xxxx xxxx	uuuu uuuu
9Eh	ADRESL	A/D Result Register Low Byte								xxxx xxxx	uuuu uuuu
1Fh	**ADCON0**	**ADCS1**	**ADCS0**	**CHS2**	**CHS1**	**CHS0**	**GO/DONE**	**—**	**ADON**	**0000 00-0**	**0000 00-0**
9Fh	**ADCON1**	**ADFM**	**ADCS2**	**—**	**—**	**PCFG3**	**PCFG2**	**PCFG1**	**PCFG0**	**00-- 0000**	**00-- 0000**
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
05h	PORTA	—	—	PORTA Data Latch when written: PORTA pins when read						--0x 0000	--0u 0000
89h(1)	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction bits			0000 -111	0000 -111
09h(1)	PORTE	—	—	—	—	—	RE2	RE1	RE0	---- -xxx	---- -uuu

BM DTMT CNĐT

Thanh ghi ADCON0

ADCS1	ADCS	CHS2	CHS1	CHS0	GO/DONE	-	ADON

Bit	Chức năng
ADCS1	ADCS0 : Lựa chọn tần số chuyển đổi
ADCS2 (ADCON1)	000= Fosc/2 001= Fosc/8 010= Fosc/32 011/111= Frc (Clock từ bộ dao động RC) 100= Fosc/4 101= Fosc/16 110= Fosc/64
CHS2-CHS1	-CHS0 : Chọn kênh Analog Input
	000 … 111= 8 trạng thái tương ứng 8 kênh vào.
GO/DONE	Bit báo trạng thái quá trình chuyển đổi A/D (khi ADON=1)
	1= Quá trình đang xảy ra hoặc Cho quá trình xảy ra. 0= Quá trình không xảy ra hoặc đã Hoàn tất.
ADON	Cho phép Module ADC hoạt động
	1= Cho phép (ADC được cung cấp nguồn) 0= Không cho phép (ADC không được cấp nguồn)

Thanh ghi ADCON1

ADFM	ADCS2			PCFG3	PCFG2	PCFG1	PCFG0

Bit	Chức năng
ADFM	Lựa chọn kiểu thanh ghi chứa kết quả chuyển đổi 10bit
	0= 10 bit trá **1**= 10 bit ph
PCFG3	PCFG0
	Cấu hình cl các chân A

PCFG <3:0>	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0
0000	A	A	A	A	A	A	A	A
0001	A	A	A	A	VREF+	A	A	A
0010	D	D	D	A	A	A	A	A
0011	D	D	D	A	VREF+	A	A	A
0100	D	D	D	D	A	D	A	A
0101	D	D	D	D	VREF+	D	A	A
011x	D	D	D	D	D	D	D	D
1000	A	A	A	A	VREF+	VREF-	A	A
1001	D	D	A	A	A	A	A	A
1010	D	D	A	A	VREF+	A	A	A
1011	D	D	A	A	VREF+	VREF-	A	A
1100	D	D	D	A	VREF+	VREF-	A	A
1101	D	D	D	D	VREF+	VREF-	A	A
1110	D	D	D	D	D	D	D	A
1111	D	D	D	D	VREF+	VREF-	D	A

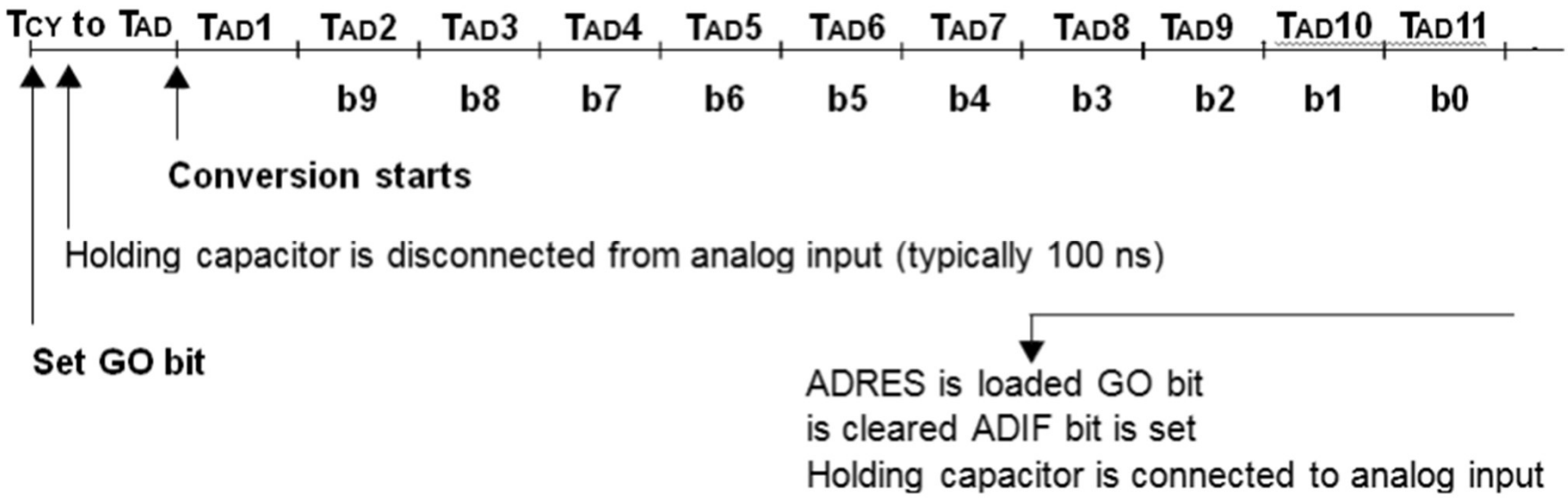
BM DTMT CNĐT

Chú ý

PCFG3:PCFG0: A/D Port Configuration Control bits

PCFG <3:0>	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C/R
0000	A	A	A	A	A	A	A	A	VDD	Vss	8/0
0001	A	A	A	A	VREF+	A	A	A	AN3	Vss	7/1
0010	D	D	D	A	A	A	A	A	VDD	Vss	5/0
0011	D	D	D	A	VREF+	A	A	A	AN3	Vss	4/1
0100	D	D	D	D	A	D	A	A	VDD	Vss	3/0
0101	D	D	D	D	VREF+	D	A	A	AN3	Vss	2/1
011x	D	D	D	D	D	D	D	D	—	—	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	AN3	AN2	6/2
1001	D	D	A	A	A	A	A	A	VDD	Vss	6/0
1010	D	D	A	A	VREF+	A	A	A	AN3	Vss	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	AN3	AN2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	AN3	AN2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	AN3	AN2	2/2
1110	D	D	D	D	D	D	D	A	VDD	Vss	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	AN3	AN2	1/2

A/D CONVERSION T_{AD} CYCLES



Refer for setting the ADC clock

AD Clock Source (TAD)		Maximum Device Frequency
Operation	ADCS2:ADCS1:ADCS0	
2 Tosc	000	1.25 MHz
4 Tosc	100	2.5 MHz
8 Tosc	001	5 MHz
16 Tosc	101	10 MHz
32 Tosc	010	20 MHz
64 Tosc	110	20 MHz
RC(1, 2)	x11	**(Note 1)**

Note 1: The RC source has a typical TAD time of 4 µs but can vary between 2-6 µs.

2: When the device frequencies are greater than 1 MHz, the RC A/D conversion clock source is only recommended for Sleep operation.

Sử dụng ngắt ADC

- Khối ADC cũng có thể tạo ra sự kiện ngắt (ngắt trong) , sự kiện ngắt xảy ra khi bộ ADC chuyển đổi hoàn tất .

Các bít:

- ADIE : Bit cho phép bộ chuyển đổi ADC.
- ADIF : Cờ ngắt ADC , bit này tự động bằng 1 khi bộ ADC chuyển đổi hoàn tất , để cho lần chuyển đổi tiếp theo được thực hiện , chúng ta cần phải xóa bít này bằng phầm mềm lập trình.
- PEIE : Bit cho phép ngắt ngoại vi .
- GIE : Bit cho phép ngắt toàn cục.

BM DTMT CNĐT

Analog to Digital Converter

1. Port configuration
2. Channel selection
3. ADC voltage reference selection
4. ADC conversion clock source
5. Interrupt control
6. Results formatting
7 . Read ADC

BM DTMT CNĐT

Programming the A/D module consists of the following steps

1. Configure the analog/digital I/O ADCON1|ANSEL/ANSELH.

   All analog lines are initialized as input (TRIS registers).
2. Configure for ADC Module
   - Select the A/D input channel (CHSx).
   - Select right- or left-justification (ADFM bit),
   - Select the reference voltage VCFGx/PCFGx
   - Select the A/D conversion clock (ADCS bits),
   - Enable the A/D module (ADON bit).
3. Configure for AD interrupt ADIF; ADIE, PEIE, GIE
4. Wait the acquisition time (t_{ACQ})
5. Initiate the conversion (GO/DONE).
6. Wait for the conversion to complete.

   polling – check GO/DONE; interrupt – check ADIF
7. Read and store the digital result (ADRESH|ADRESL

Các bước chuyển đổi từ A sang D

1. Cấu hình PORT: Chọn pin của PIC là ngõ vào (TRISXi)
   pin của PIC là analog (ANSEL, ANSELH)/ ADCON1
2. Cấu hình mô dul ADC: - Chọn kênh vào ADC (SHx).
   - Chọn điện áp mẫu tham chiếu (VCFGx/ PCFGX)
   - Chọn tần số xung chuyển đổi clock (ADCSx).
   - Định dạng dữ liệu (ADFM)
   - Cho phép bộ chuyển đổi AD hoạt động (ADON).
3. Cấu hình ngắt ADC (tùy chọn)
   Xóa cờ ngắt ADIF. Cho phép ngắt(ADIE; PEIE; GIE).
4. Đợi cho tới khi quá trình lấy mẫuổn định (t_{ACQ}=20us) .
5. Cho phép quá trình chuyrnt đổi (GO/DONE).
6. Đợi cho tới khi quá trình chuyển đổi hoàn tất:
   Polling -- > Kiểm tra bit GO/DONE .
   Ngắt -- > Kiểm tra xử lý ISR (ADIF)
7. Đọc kết quả chuyển đổi ADC (ADRESH/ADRESL)
8. Xóa cờ ngắt / và lặp lại từ bước 5 cho quá trình chuyển đổi tiếp theo

Một số bước thông dụng

Bước 2:Khởi tạo khối ADC

- **Chọn tần số chuyển đổi**

16F887	16F877A
ADCS1=....;	ADCS2:.....;
ADCS0=....;	ADCS1=.....;
	ADCS0=.....;

- **Chọn điện áp tham chiếu (ADCON1)**

16F887 <5:4>	16F877A <3:0>
VCFG1=.....;	PCFG3=.....;
VCFG0=.....;	PCFG2=.....;
	PCFG1=.....;
	PCFG0=.....;

Một số bước thông dụng

Bước 2:Khởi tạo khối ADC

- **Chọn kênh cần đo:**

  16F887 ADCON0<5:2>

  CHS3=....;CHS2=.....;CHS1=.....;CHS0=.....;

  16F877A ADCON0<5:3>

  CHS2=.....;CHS1=.....;CHS0=.....;

- **Chọn định dạng kết quả**

  ADFM=.....;

- **Cho phép module ADC**

  ADON=1;

Một số bước thông dụng

Bước 3: Khởi tạo ngắt ADC(tùy chọn):

Xóa cờ ngắt: ADIF=0;

Cho phép ngắt ADC: ADIE=1;

Cho phép ngắt ngoại vi: PEIE=1;

- Cho phép ngắt toàn cục : GIE=1;

Bước 4: Chờ thời gian khởi tạo.

t_{ACQ}= 20us

Bước 5: Bắt đầu cho phép chuyển đổi .

GODONE=1;

Một số bước thông dụng

Bước 6:Chờ cho bộ ADC chuyển đổi hoàn tất:

- Polling -- > Kiểm tra Bit GODONE = 0;
- Interrupt -- > kiểm tra ADIF=1 (bước 3)

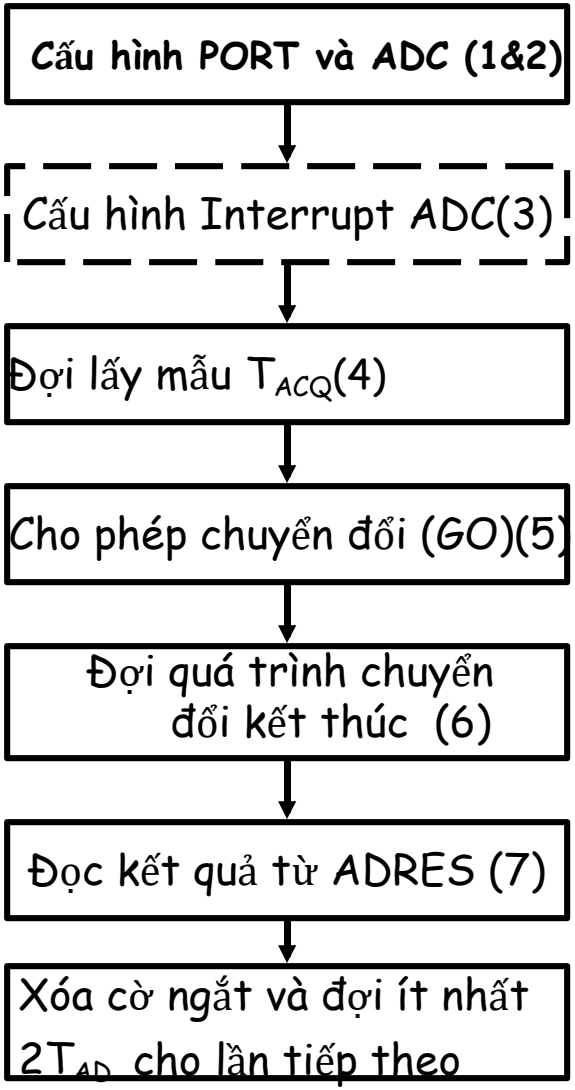
Bước 7 : Đọc kết quả

-ADC 8-bit : kết quả = ADRESL //canh phải

- ADC 10-bit : kết quả = ADRESH*64+ADRESL //canh phải

Bước 8:Xóa cờ ngắt cho lần chuyển đổi tiếp theo (Nếu bước 3 được thực hiện) ADIF=0; thực hiện từ bước 5

khởi tạo ADC bằng lưu đồ giải thuật



BÀI TẬP

Ví dụ 1:

Viết đoạn chương trình khởi tạo điều khiển ADC đọc tín hiệu tương tự (0-5v) đưa đến đầu vào AN0, định dạng dữ liệu căn phải, điện áp tham chiếu Vref+=5V; Vref-=0V, tần số chuyển đổi chọn chế độ RC, có sử dụng ngắt. Píc 16F877A

Các tham số:

Chương trình dạng ASM có dạng

Đoạn CT ASM - Khởi tạo chuyển đổi ADC

```
BSF     STATUS, RP0   ; Chọn Bank1
CLRF    ADCON1        ; Cấu hình tất cả ngỏ vào Analog
                      ; 6 bit thấp của thanh ghi kết quả = 0
BSF     PIE1, ADIE    ; Cho phép ngắt ADC.
BCF     STATUS, RP0   ; Chọn Bank0
MOVLW   b‘11000001’   ; Chọn chế độ xung RC, bit A/D được bật lên,
    ; kenh 0 được chọn, cấp nguồn cho module diện ap so sánh là Vdd và Vss.
MOVWF   ADCON0        ;
BCF     PIR1, ADIF    ; Xoá cờ ngắt ADC
BSF     INTCON, PEIE  ; Cho phép toàn bộ các ngắt ngoại vi
BSF     INTCON, GIE   ; Cho phép toàn bộ các ngắt
CALL    DELAYT_ACQ ;  Bảo đảm thời gian để lấy mẫu được giá trị chính xác.
BSF     ADCON0, GO ; Bắt đầu chuyển đổi Khi quá trình chuyển
     ; đổi hoàn tất cờ ADIF sẽ được bật lên và bit go/done được xóa đi
```

BÀI TẬP TẠI LỚP

Lập trình điều khiển sử dụng doc ADC tu dau vao AN1(RA1), Vcc=5V, toc do chuyen doi 500K(fosc/8), fosc=4M.

Kết quả đọc về (8bit cao luu trong ADRESH)

Tham số và các thanh ghi

Lưu đồ

Chương trình

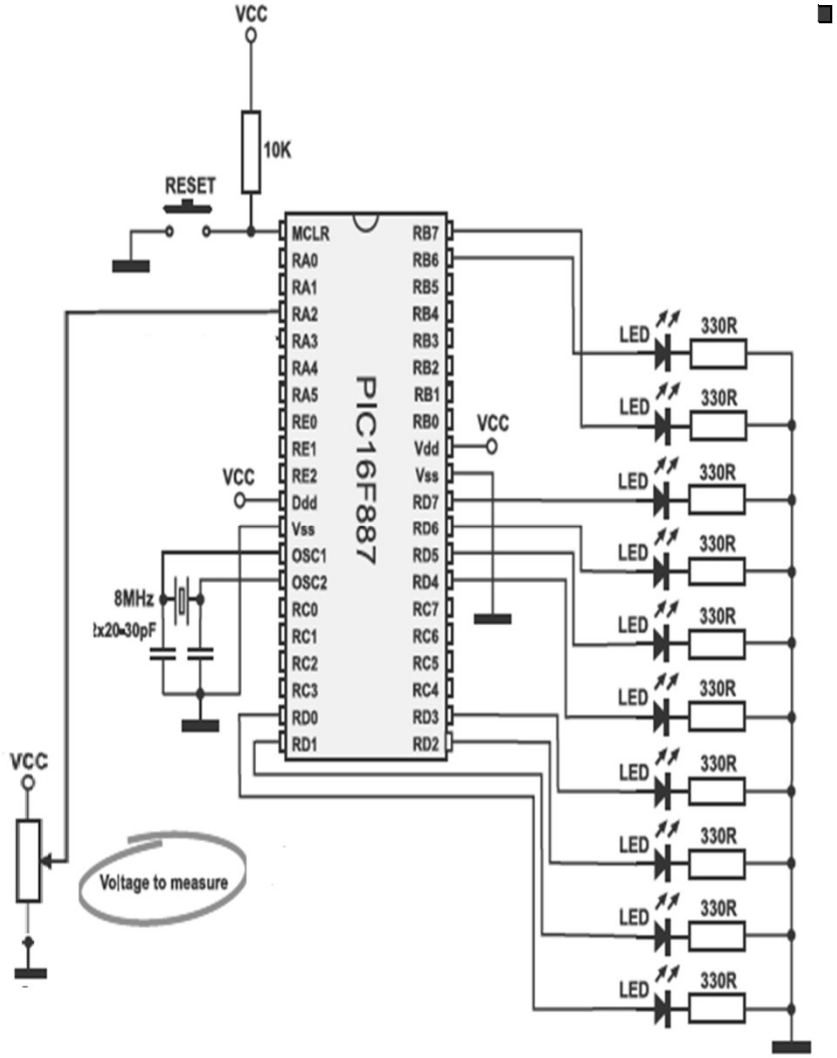
```
list   p=16f877a   ; list directive to define processor
#include    <p16f877a.inc>
 __CONFIG _CP_OFF & _WDT_OFF &
_BODEN_OFF & _PWRTE_ON & _RC_OSC &
_WRT_OFF & _LVP_ON & _
```

```
ORG   0000h
goto MAIN
MAIN
doc_adc
;chon sô dau vao ADCON1
        bsf STATUS,5
        bcf     STATUS,6
        bcf     ADCON1,3
        bsf     ADCON1,2
        bcf     ADCON1,1
        bcf     ADCON1,0
;chon dau vao (ADCON0)
        bcf     STATUS,5
        bcf     ADCON0,5
        bcf     ADCON0,4
        bsf     ADCON0,3
;
```

```
;chon tan sô lay mau
        bcf     ADCON0,7
        bsf     ADCON0,6
        bsf STATUS,5
        bcf     ADCON1,6
; chon noi luu ket qua
        bcf     ADCON1,7
        bcf     STATUS,5
; cho phep ADC làm viec
        bsf     ADCON0,2
        bsf     ADCON0,0
;kiem tra chuyen doi xong
chua?(adcon0,2 or ADIF=1?)
Loop
        btfsc ADCON0,2
        goto Loop
END
```

Bài tập ADC1

- Thiết kế mạch điện dùng vi điều khiển có chức năng đọc giá trị analog từ biến trở Voltage to measure và chuyển giá trị analog sang digital hiển thị trên 10 led đơn. Cho biết biến trở có thể điều chỉnh giá trị analog trong khoãng từ 0 đến 5V. Tần số XTAL là 4MHz.



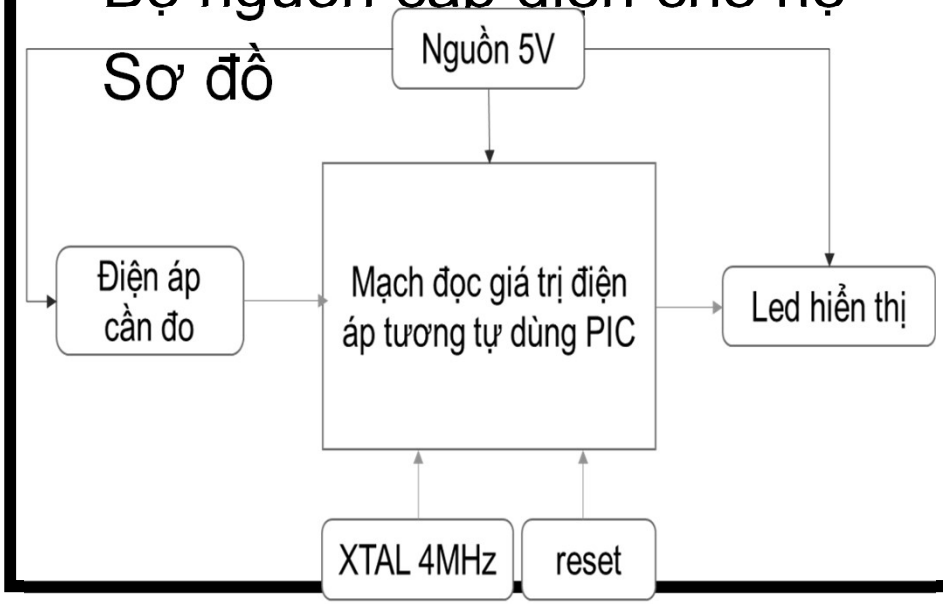
Điện áp cần đo Từ Voltage to measure

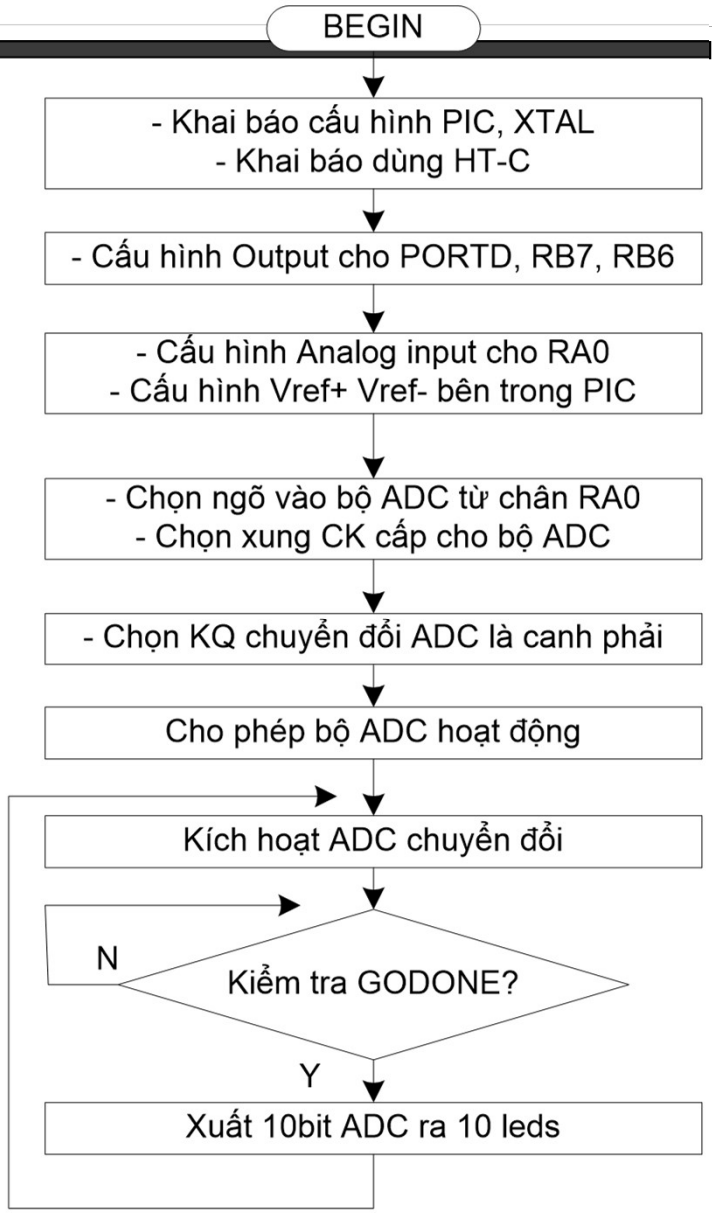
Bộ hiện thị dung 10 LED

Bộ đọc và chuyển đổi điện áp tương tự sang số dung PIC, không dung ngắt

Bộ nguồn cấp điện cho hệ

Sơ đồ





Chương trình

```
#define _XTAL_FREQ 4000000
#include <htc.h>
__CONFIG (0x20E2);
__CONFIG (0x3FFF);
 void main (void)
{
// Đối với 16f887
ANSEL=0x00; ANSELH=0x00;
ANS0=1;
 TRISD=0x00; TRISB7=0;
TRISB6=0;
 ADCS1=1; ADCS0=1;          //
RC Clock
 // Đối với 16f887
VCFG1=0; VCFG0=0;
CHS<3:0>=0B0000;
      //Đối với 16f877
 PCFG<3:0>=0B1110;
 CHS<2:0>=0B000;
ADFM=1;
ADON=1;

while (1)
{
    GODONE = 1;
     while (GODONE == 1);
     PORTD = ADRESL;
    PORTB = ADRESH <<6;
   }
}
```

Bài tập ADC2

- Thiết kế mạch điện dùng vi điều khiển có chức năng đọc giá trị giá trị analog từ biến trở đưa đến RA0 và chuyển giá trị analog sang digital hiển thị trên 10 led đơn. Cho biết biến trở có thể điều chỉnh giá trị analog trong khoảng từ 0 đến 3.3V. Điện áp tham chiếu $V_{REF(+)}$ lấy từ bên ngoài vi điều khiển đua đến RA3.

Sơ đồ xem bài 1 và sửa cho phù hợp

Điện áp cần đo từ nguồn (0-3.3v) đưa đến chân RA0,

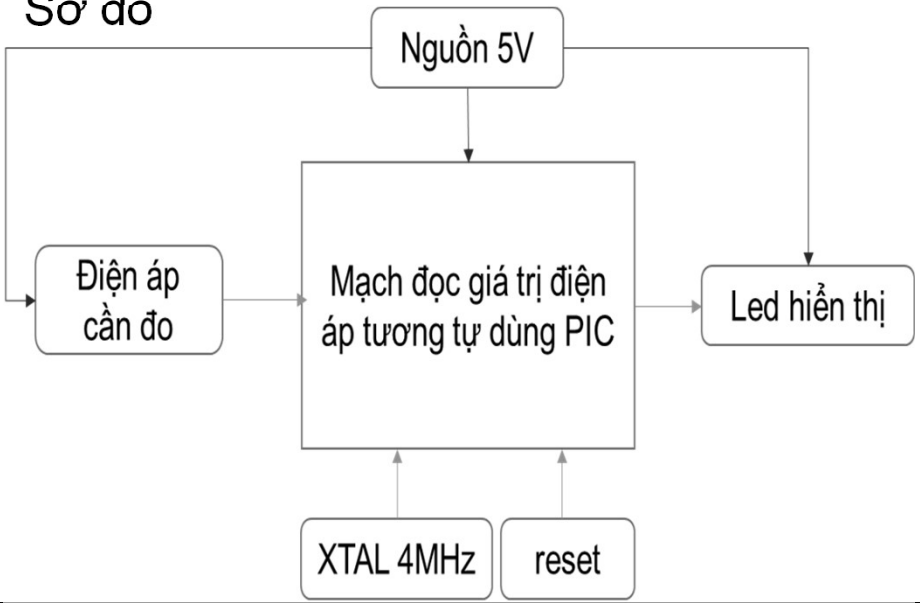
Điện áp 3,3VDC đưa đến RA3 làm V_{REF}^{+}

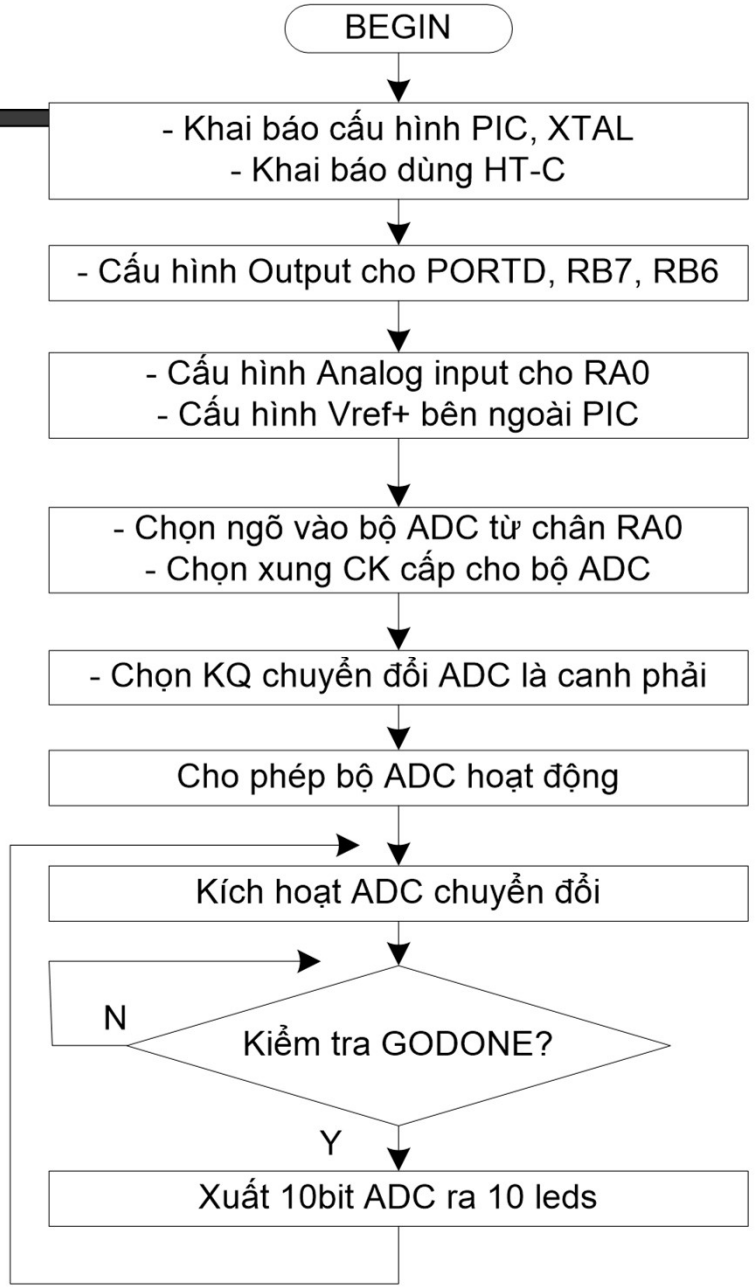
Bộ hiện thị dung 10 LED

Bộ đọc và chuyển đổi điện áp tương tự sang số dung PIC, không dung ngắt

Bộ nguồn cấp điện cho hệ

Sơ đồ





```
#define _XTAL_FREQ 4000000
#include <htc.h>
__CONFIG (0x20E2);
__CONFIG (0x3FFF);
 void main (void)
{
// Đối với 16f887
ANSEL=0x00; ANSELH=0x00;
ANS0=1;
TRISD=0x00; TRISB7=0; TRISB6=0;
 ADCS1=1; ADCS0=1;//RCClock
 // Đối với 16f887
VCFG0=1; VCFG1=0;
CHS<3:0>=0B0000;
//Đối với 16f877
PCFG<3:0>=0B0101;
CHS<2:0>=0B000;
ADFM=1;
ADON=1;
while (1)
   {
    GODONE = 1;
    while (GODONE == 1);
    PORTD = ADRESL;
   PORTB = ADRESH <<6;
   }
}
```

- Thiết kế mạch điện dùng vi điều khiển đọc giá trị analog từ biến trở đưa đến RA0 và chuyển giá trị analog sang digital hiển thị trên 10 led đơn. Cho biết giá trị analog nằm trong khoãng từ 1.25V đến 3.3V. Điện áp tham chiếu lấy từ bên ngoài vi điều khiển: $V_{REF(+)}$ đưa đến RA3; $V_{REF(-)}$ đưa đến RA2.
- Sơ đồ nguyên lý- mô phỏng

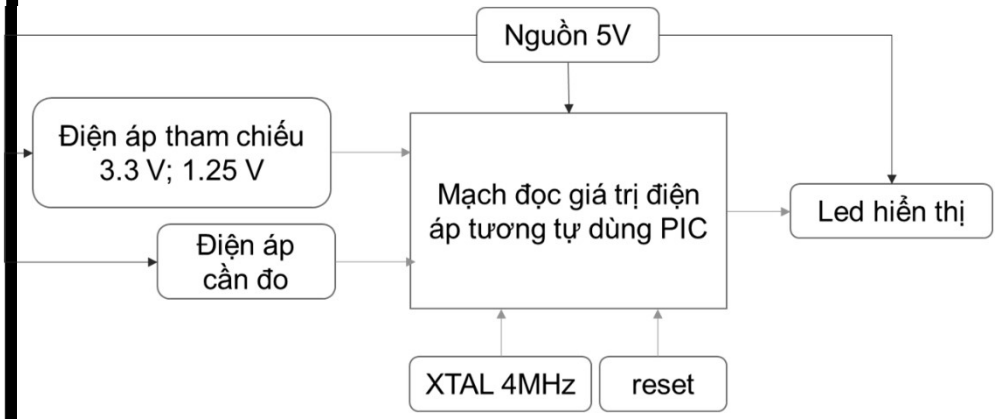
Điện áp cần đo từ nguồn (1,25 -3.3v) đưa đến chân RA0,

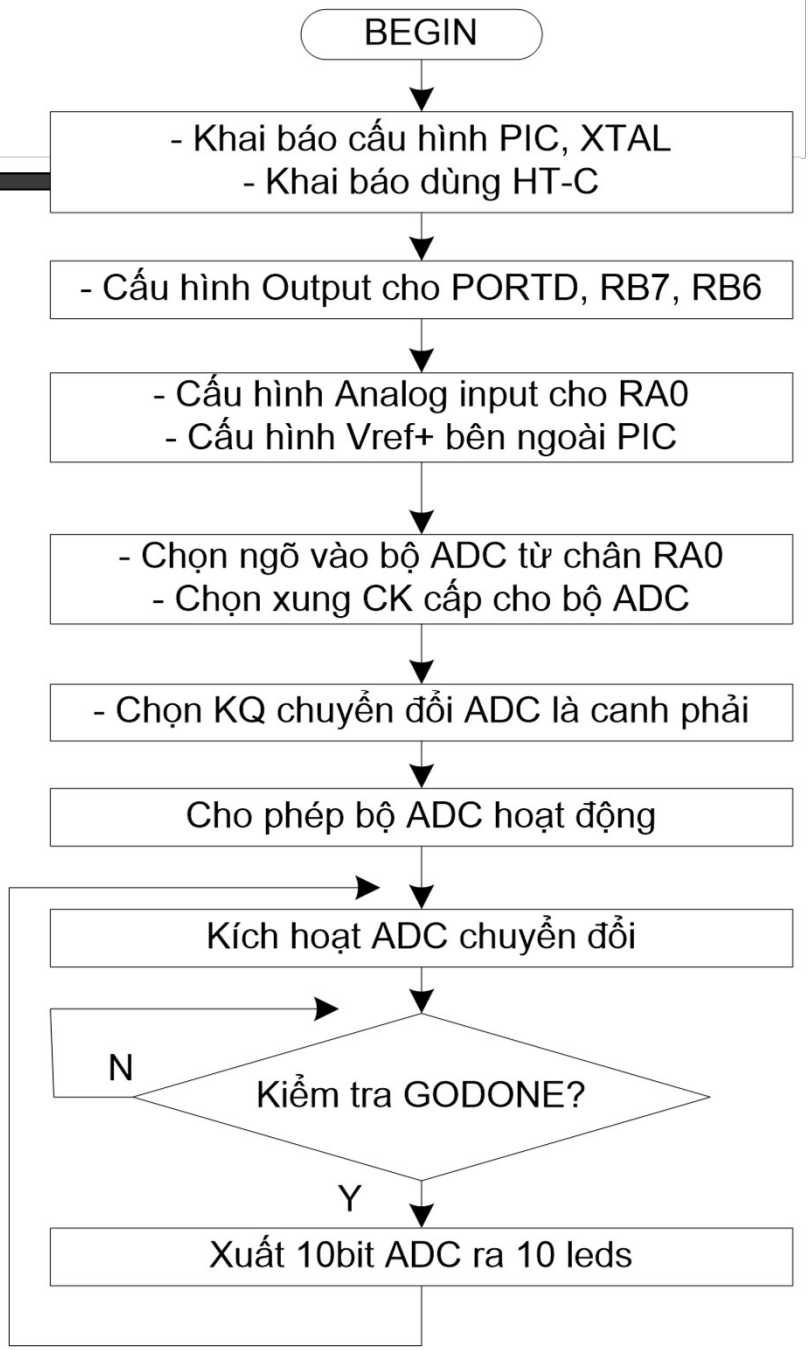
Điện áp 3,3VDC đưa đến RA3 làm V_{REF}^{+}; điện áp 1,25 VDC đưa đến RA2 làm V_{REF-}

Bộ hiện thị dung 10 LED

Bộ đọc và chuyển đổi điện áp tương tự sang số dung PIC, không dung ngắt

Sơ đồ





```
#define _XTAL_FREQ 4000000
#include <htc.h>
__CONFIG (0x20E2);
__CONFIG (0x3FFF);
 void main (void)
{
// Đối với 16f887
ANSEL=0x00; ANSELH=0x00;
ANS0=1;
 TRISD=0x00; TRISB7=0; TRISB6=0;
 ADCS1=1; ADCS0=1;       // RC
Clock
 // Đối với 16f887
VCFG0=1; VCFG1=1;
CHS<3:0>=0B0000;
 //Đối với 16f877
PCFG<3:0>=0B0101;
CHS<2:0>=0B000;
ADFM=1;
ADON=1;
while (1)
{
GODONE = 1;
while (GODONE == 1);
PORTD = ADRESL;
PORTB = ADRESH <<6;
}
}
```