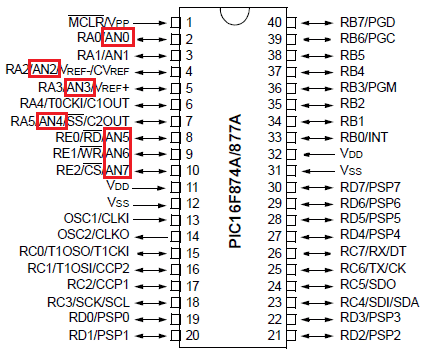
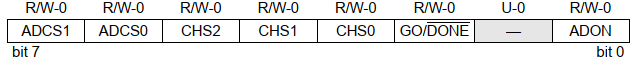
1. **Tổng quan về ADC trong PIC16F877A**

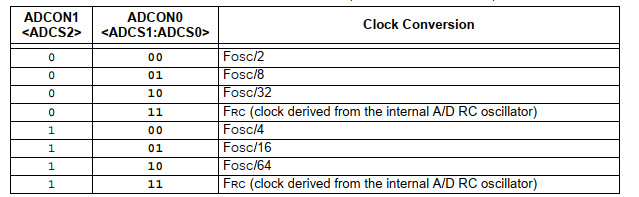
* PIC có 10 bit, 8 channel ADC, điều này đồng nghĩa với giá trị đầu ra sẽ từ 0 – 1024 (2^10) và có 8 pins (8 channels) trên MCU có thể đọc tín hiệu analog
* Trước khi muốn sử dụng channel nào, ta phải cấu hình trong code
* A/D module có 4 thanh ghi, chúng phải được cấu hình để đọc dữ liệu từ các chân đầu vào:
  + A/D Result High Register (ADRESH)
  + A/D Result Low Register (ADRESL)
  + A/D Control Register 0 (ADCON0)
  + A/D Control Register 1 (ADCON1)
* Thanh ghi ADCON0 điều khiển hoạt động của A/D module
* Thanh ghi ADCON1 điều dùng để cấu hình các chức năng của các chân

1. **Các thanh ghi của ADC**

**A, A/D Control Register 0 (ADCON0)**



Bit 7 – 6(ADCS1 – ADCS0): các bit chọn clock cho ADC:



Bit 5 – 3(CHS2 – CHS0): các bit chọn channel

000 = Channel 0 (AN0)  
001 = Channel 1 (AN1)  
010 = Channel 2 (AN2)  
011 = Channel 3 (AN3)  
100 = Channel 4 (AN4)  
101 = Channel 5 (AN5)  
110 = Channel 6 (AN6)  
111 = Channel 7 (AN7)

Bit 2: GO/DONE: bit bắt đầu chuyển đổi A/D

* Khi bit ADON = 1:

+, Go/done = 1 thì A/D được chuyển đổi

+, Go/done = 0 thì A/D không được chuyển đổi

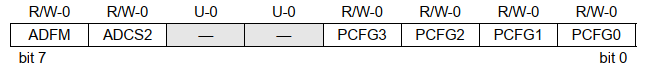
Bit 1: chưa thực hiện, set ở mức 0

Bit 0: ADON bit (AD on)

* ADON = 1: ADC module được cấp nguồn
* ADON = 0: ADC module tắt
* Ta sẽ config thanh ghi ADCON0 như sau: ADCON0 = 0x41 (0100 0001)

//cấp nguồn cho module ADC, lựa chọn clock là Fosc/16

**B, A/D Control Register 1 (ADCON1)**

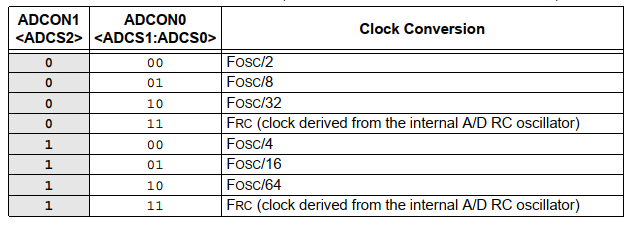


Bit 7: ADFM (A/D result format select bit)

1 = kết quả sẽ dịch phải

0 = kết quả sẽ dịch trái

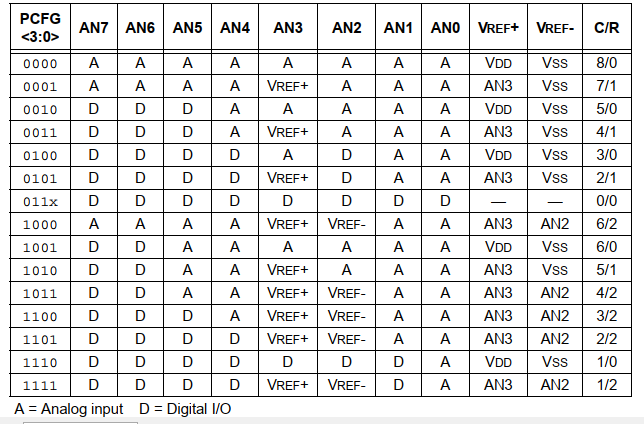
Bit 6: ADCS2



Ta sẽ cấu hình bit này sao cho clock bằng với clock đã chọn ở thanh ghi ADCON0

Bit 5-4: set ở mức 0

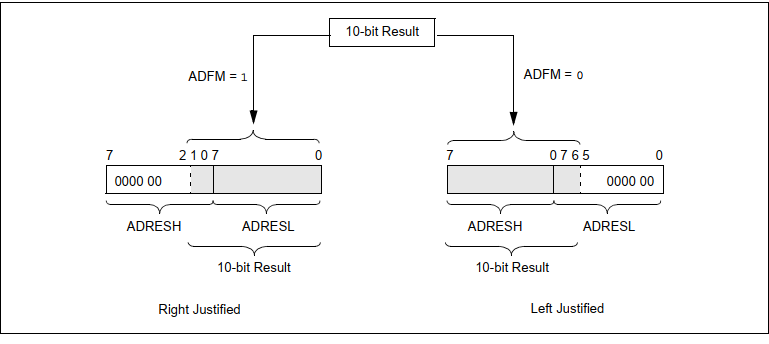
Bit 3-0: các bit điều khiển cấu hình cho cổng A/D



* Ta sẽ cấu hình cho thanh ghi ADCON1 như sau: ADCON1 = 0xCB

//(1100 1011), cấu hình để lưu kết quả dịch phải trên 2 thanh ghi ADRESH-ADRESL. Chọn clock = Fosc/16. Chọn AN3 và AN2 là 2 chân Vref+, Vref-

**C, Hai thanh ghi ADRESH và ADRESL**



Như đã trình bày ở phần thanh ghi ADCON1, ta sẽ chọn định dạng lưu kết quả nhận được bằng cách set bit ADFM bằng 1 hoặc 0, để 10bit kết quả nhận về sẽ được lưu trên 2 thanh ghi ADRESH và ADRESL dịch trái hay dịch phải

**3, Code các hàm cho ADC**

**A, Khởi tạo ADC**

void ADC\_Init**(**void**)**

**{**

ADCON0 **=** 0x41**;** // config ADCON0: ADC mode: ON, with Fosc/16

ADCON1 **=** 0xCB**;** // 1100 1011, AN3(+), AN2(-) set to Vref, Fosc/16

**}**

Cấu hình 2 thanh ghi ADCON0 và ADCON1 như đã trình bày ở mục 2 A,B

**B, Hàm đọc tín hiệu**

unsigned int ADC\_ReadChannel**(**unsigned int channel**)**

**{**

ADCON0 **&=** 0xC5**;** // 1100 0101, clear bit 5-3(analog channel select bit)

ADCON0 **|=** channel **<<** 3**;** //setting bit bit 5-3(analog channel select bit)

Delay\_ms**(**2**);** //time hold capacitor

GO\_DONE\_bit **=** 1**;** // Go/DONE bit high(1), A/D conversion in progress

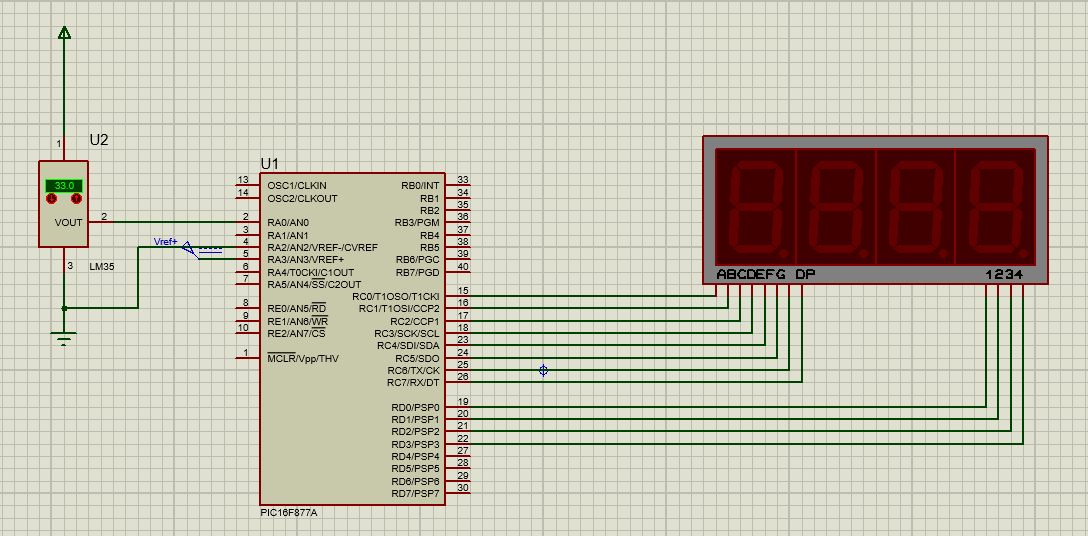
**while(**GO\_DONE\_bit**);** //wait A/D conversion complete, after completed GO/DONE bit will low(0)

**return** **((**ADRESH**<<**8**)+**ADRESL**);** //Returns Result

}

**4, Đọc tín hiệu từ LM35**

A, Sơ đồ nguyên lý



* Kết nối: chân output của LM32 với pin RA0/AN0
* Ta sẽ nối chân RA3 (Vref+) với 1 nguồn DC 10V và RA2 (Vref-) với GND
* Sử dụng PortC và PortD để kết nối với led 7 thanh

B, Code

void main**()**

**{**

unsigned int adc\_input**;**

float volt**,** temp**,** temp\_display**;**

TRISC **=** 0x00**;** //port C set as output

TRISD **=** 0xF0**;** //pin 0, 1, 2, 3 of port D set as output

PORTC **=** 0xFF**;** //off led 7seg

ADC\_Init**();** //adc intilaze

**while(**1**)**

**{**

adc\_input **=** ADC\_ReadChannel**(**0**);**

volt **=** adc\_input **\*** 10000.0f **/** 1023**;**

temp **=** volt**/** 10.0f**;**

temp\_display **=** temp **\*** 100**;**

**for(**i **=** 0**;** i **<=** 125**;** i**++)**

Led7\_Quartet**(**temp\_display**);**

**}**

**}**