Electronics 372 an Introduction to Microprocessors
from datasheet and online resources

This will contain several versions of explanations.
My notes are at the start,

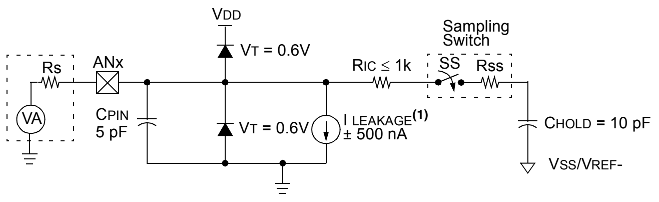
other notes are added from
http://electronicseverywhere.blogspot.com/2010/10/pic16f887877-programming-in-c-tutorial.html

at the end is the material provided by mikroelectronica
http://www.mikroe.com/chapters/view/16/chapter-3-pic16f887-microcontroller/#c3v9

The ADC converts by moving the voltage to single sample and hold circuit. Since there are up to 14 inputs these are multiplexed to this holding circuit. The multiplexer is shown. The holding circuit is part of the ADC block.

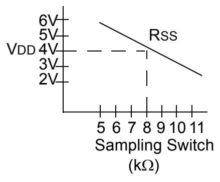
Discussion of the sample and hold circuit and the time required to get and hold the voltage.
"A typical sample and hold circuit stores electric charge in a capacitor and contains at least one fast FET switch and at least one operational amplifier.[1] To sample the input signal the switch connects the capacitor to the output of a buffer amplifier. The buffer amplifier charges or discharges the capacitor so that the voltage across the capacitor is practically equal, or proportional to, input voltage. In hold mode the switch disconnects the capacitor from the buffer. The capacitor is invariably discharged by its own leakage currents and useful load currents, which makes the circuit inherently volatile, but the loss of voltage (*voltage drop*) within a specified hold time remains within an acceptable error margin." wiki excerpt
For the ADC to meet its specified accuracy, the charge holding capacitor (C_{HOLD}) must be allowed to fully charge to the input channel voltage level. The Analog Input model is shown in Figure9-4. The source impedance (R_S) and the internal sampling switch (R_{SS}) impedance directly affect the time required to charge the capacitor C_{HOLD}. The sampling switch (R_{SS}) impedance varies over the device voltage (V_{DD}), see Figure 9-4. Note that the ANx is the source of the signal to be measured $V_{applied}$. It will charge up the capacitor C_{HOLD} when the sampling switch is closed. How long until the capacitor will charge to the requisite voltage V_{hold} ? Since capacitors charge exponentially the final voltage reaches the applied voltage at $t=\infty$ but the voltage V_{hold} need only be within one significant digit of the applied voltage. $V_{applied} - V_{hold} <$ ADC resolution then you are ok.

FIGURE 9-4: ANALOG INPUT MODEL



Legend:

CPIN	= Input Capacitance
VT	= Threshold Voltage
I LEAKAGE	= Leakage current at the pin due to various junctions
RIC	= Interconnect Resistance
SS	= Sampling Switch
CHOLD	= Sample/Hold Capacitance



Note 1: See **Section 17.0 "Electrical Specifications"**.

The maximum recommended impedance for analog sources R_S is 10 kΩ. This represents the resistance added by the voltage source. In principal this relates the voltage at the pin and the current that can be delivered by the source. Consider that if there is 5V applied and there is (10+5) pF to be charged then

VC=q 5*15= 75 pC

75 pC of charge must be delivered by the source before the pin reaches the voltage. Initially with a 10 kΩ a V/R=5/10,000 = 0.5 mA current can be applied to the pin. This initial current could deliver the 75 pC in $75\text{x}10^{-12}/5\text{x}10^{-4} = 15\text{x}10^{-8}$=150ns. However the relationship that determines the time is the charging of the 15 pF via the exponential law. The time constant for this is (neglect the 5pF input cap.)

RC=CHOLD(RIC + RSS + RS) = 10pF(1+10+10) kΩ=210 nS and you need to charge to about 0.1% (10 bit resolution)

$V_{hold}/V_{applied}=0.001=e^{(-t/RC)}$

ln(1/1000)=t/RC

t= 7(RC)= 1.4 µs.

So you need a few µs to establish the voltage in the sample and hold circuit. The actual time

TACQ = Amplifier Settling Time + Hold Capacitor Charging Time + Temperature Coefficient

is about 5 µs.

from the datasheet
In order to enable the ADC to meet its specified accuracy, it is necessary to provide a certain time delay between selecting specific analog input and measurement itself. This time is called 'acquisition time' and mainly depends on the source impedance. There is an equation used to calculate this time accurately, which in the worst case amounts to approximately 20uS. So, if you want the

conversion to be accurate, don't forget this important detail.

A sample calculation of the time from the datasheet is shown below. Note the units are [us or Ms] micro seconds.

Assumptions: *Temperature = 50°C and external impedance of 10kΩ 5.0V VDD*

$$\begin{aligned} T_{ACQ} &= \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient} \\ &= T_{AMP} + T_C + T_{COFF} \\ &= 2\mu s + T_C + [(\text{Temperature} - 25°C)(0.05\mu s/°C)] \end{aligned}$$

The value for TC can be approximated with the following equations:

$$V_{APPLIED}\left(1 - \frac{1}{(2^{n+1}) - 1}\right) = V_{CHOLD} \qquad \text{;[1] } V_{CHOLD} \text{ charged to within 1/2 lsb}$$

$$V_{APPLIED}\left(1 - e^{\frac{-Tc}{RC}}\right) = V_{CHOLD} \qquad \text{;[2] } V_{CHOLD} \text{ charge response to } V_{APPLIED}$$

$$V_{APPLIED}\left(1 - e^{\frac{-Tc}{RC}}\right) = V_{APPLIED}\left(1 - \frac{1}{(2^{n+1}) - 1}\right) \qquad \text{;combining [1] and [2]}$$

Solving for TC:

$$\begin{aligned} T_C &= -C_{HOLD}(R_{IC} + R_{SS} + R_S)\ln(1/2047) \\ &= -10pF(1k\Omega + 7k\Omega + 10k\Omega)\ln(0.0004885) \\ &= 1.37\mu s \end{aligned}$$

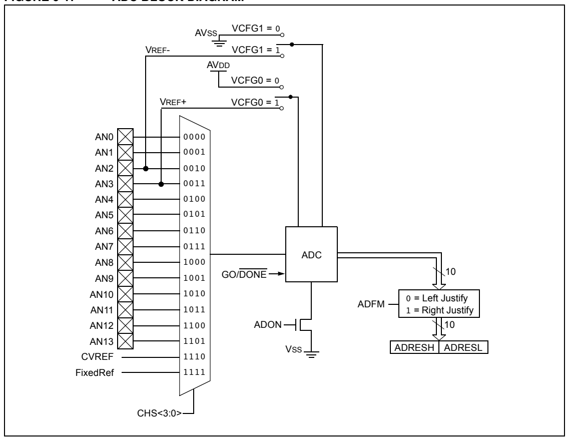
Therefore:

$$\begin{aligned} T_{ACQ} &= 2MS + 1.37MS + [(50°C - 25°C)(0.05MS/°C)] \\ &= 4.67MS \end{aligned}$$

end of sample and hold discussion

The circuit below shows multiplexing. All the signals on the left are potential inputs to ADC. Also pins AN2 and AN3 can be chosen as the voltage references.

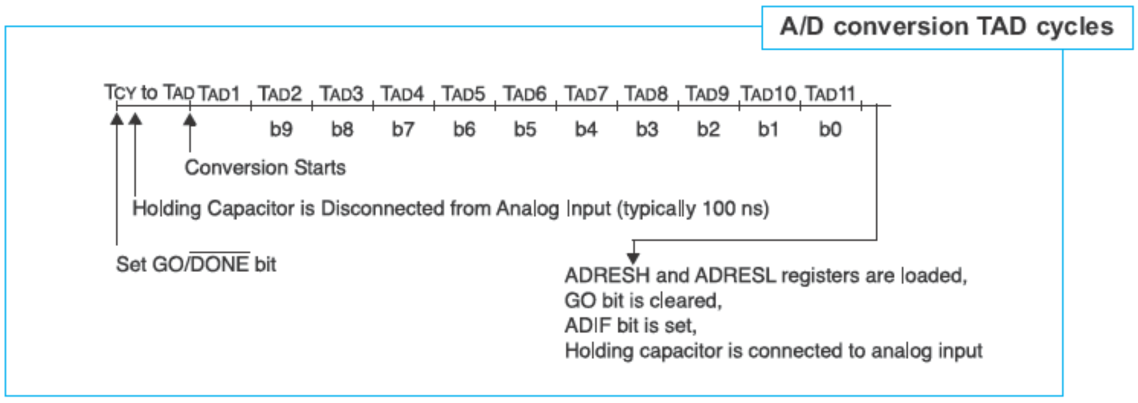
FIGURE 9-1: ADC BLOCK DIAGRAM



In addition to the analog channels CVREF (comparator voltage reference), FixedRef can be measure. Note you can use AN2, AN3 as either the +- voltage references or as an analog input. As references they are switched directly to the ADC. The conversion is based on the successive approximation scheme. Voltage comparisons

are done between the held voltage and internal voltages generated by the ADC circuitry. The range that is explored is set by reference voltages. Using the internal reference is equal to VDD, not 5V. So, if VDD = 5V, VREF+ = 5V. But if VDD = 3.3V, VREF+ = 3.3V. In both cases, VSS is 0V, so VREF- is 0V. Using external reference with VREF- of 3V and VREF+ of 5V the acceptable input voltage range is 3V to 5V.
Once the voltage has stabilized the conversion can start. The conversion first find if the voltage is in the hi range or the low range [one TAD cycle]. The voltage is then compared with the hi low of the ½ where the voltage was found [TAD 2]. Each cycle is a comparison hi or low in the region. The table displays this process for a 4 cycle 4 bit digitization. Assume range is 0-5V, ff the first comparison reveals the v>2.5, second finds V<3.8 ... you reach a binary result 1001.

hi	hi	hi	hi	1111
			lo	1110
		lo	hi	1101
			lo	1100
	lo	hi	hi	1011
			lo	1010
		lo	hi	1001
			lo	1000
low	hi	hi	hi	0111
			lo	0110
		lo	hi	0101
			lo	0100
	low	hi	hi	0011
			lo	0010
		lo	hi	0001
			lo	0000


A/D conversion TAD cycles
TCY to TAD TAD1 TAD2 TAD3 TAD4 TAD5 TAD6 TAD7 TAD8 TAD9 TAD10 TAD11
b9 b8 b7 b6 b5 b4 b3 b2 b1 b0
Conversion Starts
Holding Capacitor is Disconnected from Analog Input (typically 100 ns)
Set GO/DONE bit
ADRESH and ADRESL registers are loaded,
GO bit is cleared,
ADIF bit is set,
Holding capacitor is connected to analog input

ADC Configuration --------

1. Port configuration
2. Channel selection
3. ADC voltage reference selection

4. ADC conversion clock source
5. Interrupt control
6. Results formatting

1 PORT CONFIGURATION----------
The ADC can be used to convert both analog and digital signals. When converting analog signals, the I/O pin should be configured for analog by setting the associated TRIS and ANSEL bits. See the corresponding Port section for more information.

2 CHANNEL SELECTION ----------
The CHS bits of the ADCON0 register determine which channel is connected to the sample and hold circuit.

3 ADC VOLTAGE REFERENCE-------
The VCFG bits of the ADCON0 register provide independent control of the positive and negative voltage references. The positive voltage reference can be either V_{DD} or an external voltage source. Likewise, the negative voltage reference can be either V_{SS} or an external voltage source.

4 CONVERSION CLOCK---------------
The source of the conversion clock is software select-able via the ADCS bits of the ADCON0 register. There are four possible clock options:
• FOSC/2 • FOSC/8 • FOSC/32 • FRC (dedicated internal oscillator)
The time to complete one bit conversion is defined as T_{AD}. One full 10-bit conversion requires 11 T_{AD} periods. For correct conversion, the appropriate T_{AD} specification must be met. See A/D conversion requirements in Section 17.0 “Electrical Specifications” for more information

Interrupt control:
The ADC module allows for the ability to generate an interrupt upon completion of an Analog-to-Digital conversion. The ADC interrupt flag is the ADIF bit in the PIR1 register. The ADC interrupt enable is the ADIE bit in the PIE1 register. The ADIF bit must be cleared in software.

When the conversion is complete, the ADC module will:
- CleartheGO/DONEbit
- SettheADIFflagbit
- UpdatetheADRESH:ADRESL

step 1	Configure Port: Disable pin output driver(SeeTRISregister) Configure pin as analog [use AN2]
TRIS	choose input or output choose input TRISA = oxFF; set all portA as inputs
ANSEL	choose analog or digital

	ansel=4; // Configure AN2 pin as analog
	anselh=0;

step2	Configure the ADC module: • Select ADC conversion clock • Configure voltage reference • Select ADC input channel • Select result format • Turn on ADC module
ADCON	connect input to ADC

ADCON0	ADCS1	ADCS0	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON

	The CHS bits of the ADCON0 register determine which channel is connected to the sample and hold circuit 4 bits [max # 16 14 ADC channels AN0-AN13, plus 2 others CVREF, FixedRef see multiplexing circuit, note that PIC16F887 the CVREF/VREF- are same and share also the AN2 pin] The VCFG bits of the ADCON0 register provide independent control of the positive and negative voltage references. The positive voltage reference can be either VDD or an external voltage source. Likewise, the negative voltage reference can be either VSS or an external voltage source.External [AN2, AN3] clock options: frequency and divisor 4 options via ADCS1, ADCS0 FOSC/2 [00], FOSC/8 [01], FOSC/32[10], FRC[11], (internal oscillator) Assume that FOSC is the system clock that is already configured [normally use 8MHz internal] Then you can further divide the frequency. 8/32➔ .25 MHz or 4µsec, 8/8➔1 µsec need about 1.6 µsec per tad.
ADCON1	ADFM: A/D Cnv. Result Format bit 1 = Right justified 0 = Left justified VCFG1: Voltage Reference bit 1 = VREF- pin 0 = VSS VCFG0: Voltage Reference bit 1 = VREF+ pin 0 = Vdd As shown above the AN2 and AN3 pins can be used as the reference voltages .

ADFM	—	VCFG1	VCFG0	—	—	—	—

3	enable & go must wait ~10µs between enable & go
ADON go/done	To enable the ADC module, the ADON bit of the ADCON0 register must be set to a '1'. Wait the required acquisition time(2). Start conversion by setting the GO/DONE bit. Setting the GO/ DONE bit of the ADCON0 register to a '1' will start the Analog-to-Digital conversion. When the conversion is complete, the ADC module will: • Clear GO/DONE bit • Set ADIF flagbit

	• Update registers with conversion result
ADRESH ADRESL	2 registers that contain the digital result 10 bit binary number 0-1023
	Wait for ADC conversion to complete by one of the following: • Polling the GO/DONE bit • Waiting for the ADC interrupt (interrupts enabled) Read ADC Result [Clear the ADC interrupt flag (required if interrupt is enabled)]

The time to complete one bit conversion is defined as TAD. One full 10-bit conversion requires 11 TAD periods as shown in Figure 9-2.

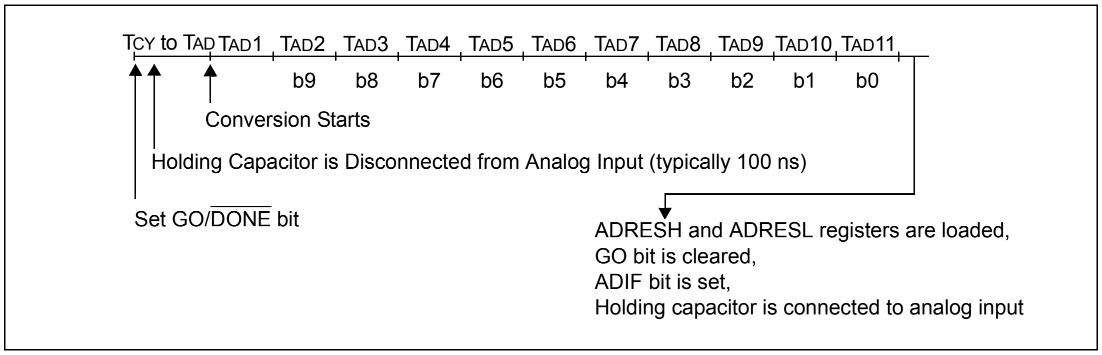
For correct conversion, the appropriate TAD specification must be met. See A/D conversion requirements in Section 17.0 "Electrical Specifications" for more information. Table9-1 gives examples of appropriate ADC clock selections.

TABLE 9-1: ADC CLOCK PERIOD (TAD) VS. DEVICE OPERATING FREQUENCIES (VDD ≥ 3.0V)

ADC Clock Period (TAD)		Device Frequency (FOSC)			
ADC Clock Source	**ADCS<1:0>**	**20 MHz**	**8 MHz**	**4 MHz**	**1 MHz**
FOSC/2	00	100 ns$^{(2)}$	250 ns$^{(2)}$	500 ns$^{(2)}$	2.0 μs
FOSC/8	01	400 ns$^{(2)}$	1.0 μs$^{(2)}$	2.0 μs	8.0 μs$^{(3)}$
FOSC/32	10	1.6 μs	4.0 μs	8.0 μs$^{(3)}$	32.0 μs$^{(3)}$
FRC	11	2-6 μs$^{(1,4)}$	2-6 μs$^{(1,4)}$	2-6 μs$^{(1,4)}$	2-6 μs$^{(1,4)}$

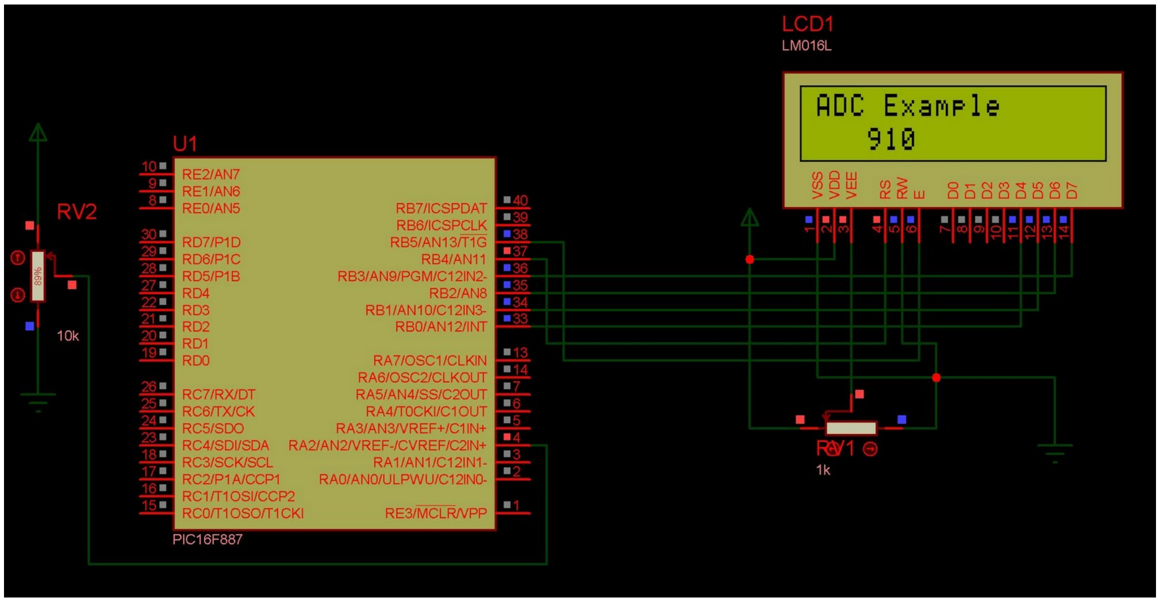
Legend: Shaded cells are outside of recommended range.
Note 1: The FRC source has a typical TAD time of 4 μs for VDD > 3.0V.
2: These values violate the minimum required TAD time.
3: For faster conversion times, the selection of another clock source is recommended.
4: When the device frequency is greater than 1 MHz, the FRC clock source is only recommended if the conversion will be performed during Sleep.

FIGURE 9-2: ANALOG-TO-DIGITAL CONVERSION TAD CYCLES


TCY to TAD
TAD1
TAD2
TAD3
TAD4
TAD5
TAD6
TAD7
TAD8
TAD9
TAD10
TAD11
b9
b8
b7
b6
b5
b4
b3
b2
b1
b0
Conversion Starts
Holding Capacitor is Disconnected from Analog Input (typically 100 ns)
Set GO/DONE bit
ADRESH and ADRESL registers are loaded,
GO bit is cleared,
ADIF bit is set,
Holding capacitor is connected to analog input

see

http://electronicseverywhere.blogspot.com/2010/10/pic16f887877-programming-in-c-tutorial.html



The discussion of the LCD programming is in LCD.doc or LCD.pdf
To test the conversion you can set up a 10k pot {RV2} which can be used to dial in 0 to 5V signal to input into the analog input [AN2].

Analog to Digital Converter (ADC):
The Analog-to-Digital Converter (ADC)
analog input signal ➔ 10-bit binary analog inputs {AN2}➔ multiplexed {AN0-AN13} into a single sample and hold circuit ➔ converter ➔conversion ➔registers (ADRESL and ADRESH).

voltage reference is selectable internal or externally supplied.

A/D CONVERSION PROCEDURE:
This is an example procedure for using the ADC to perform an Analog-to-Digital conversion:
1. Configure Port:
• Disable pin output driver (See TRIS register)
• Configure pin as analog
2. Configure the ADC module:
• Select ADC conversion clock
• Configure voltage reference
• Select ADC input channel
• Select result format
• Turn on ADC module
3. Configure ADC interrupt (optional):
• Clear ADC interrupt flag
• Enable ADC interrupt
• Enable peripheral interrupt
• Enable global interrupt(1)
4. Wait the required acquisition time(2).
5. Start conversion by setting the GO/DONE bit.
6. Wait for ADC conversion to complete by one of the following:
• Polling the GO/DONE bit
• Waiting for the ADC interrupt (interrupts enabled)
7. Read ADC Result
8. Clear the ADC interrupt flag (required if interrupt is enabled).

Code:

ADC LIBRARY

ADC (Analog to Digital Converter) module is available with a number of PIC MCU models. Library function `ADC_Read` is included to provide you comfortable work with the module.

ADC_Read

Prototype	`unsigned ADC_Read(unsigned short channel);`
Returns	10-bit unsigned value read from the specified channel.
Description	Initializes PIC's internal ADC module to work with RC clock. Clock de performing AD conversion (min 12TAD). Parameter `channel` represents the channel from which the analog v appropriate datasheet for channel-to-pin mapping.
Requires	Nothing.
Example	`unsigned tmp; ... tmp = ADC_Read(2); // Read analog value`

Library Example

This example code reads analog value from channel 2 and displays it on PORTB and PORTC.

```
unsigned int temp_res;

void main() {
// Configure AN2 pin as analog
// PORTA is input
ANSEL = 0x04;
TRISA = 0xFF;
// Configure other AN pins as digital I/O
ANSELH = 0;
// Pins RC7, RC6 are outputs
// PORTB is output
TRISC = 0x3F;
TRISB = 0;
// Get 10-bit results of AD conversion
// Send lower 8 bits to PORTB
// Send 2 most significant bits to RC7, RC6
        do {
        temp_res = ADC_Read(2);
        PORTB = temp_res;
        PORTC = temp_res >> 2;
        } while(1);
}
```

ANOTHER EXAMPLE
Lets make it easy by using adc library of mikroc. The following code will convert the analog input, at pin RA2, into digital and display it on lcd. As the conversion is of 10-bit, so the range is from 0-1023.

```
unsigned int temp_res;

// LCD module connections
sbit LCD_RS at RB4_bit;
sbit LCD_EN at RB5_bit;
sbit LCD_D4 at RB0_bit;
sbit LCD_D5 at RB1_bit;
sbit LCD_D6 at RB2_bit;
sbit LCD_D7 at RB3_bit;

sbit LCD_RS_Direction at TRISB4_bit;
sbit LCD_EN_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB0_bit;
sbit LCD_D5_Direction at TRISB1_bit;
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D7_Direction at TRISB3_bit;
// End LCD module connections

char txt1[] = "ADC Example";
char txt[7];
void main() {
////////// comment for 877//////////////////////////////////////
  ansel=4;                    // Configure AN2 pin as analog
  anselh=0;
  c1on_bit=0;
  c2on_bit=0;
 ////////////////////////////////////////////////////////////////
  TRISA  = 0xFF;             // PORTA is input
  Lcd_Init();
  Lcd_Cmd(_LCD_CLEAR);               // Clear display
  Lcd_Cmd(_LCD_CURSOR_OFF);          // Cursor off
  Lcd_Out(1,1,txt1);
  adc_init();
  do {

    temp_res = ADC_read(2);   // Get 10-bit results of AD conversion
    IntToStr(temp_res, txt);    //int to string conversion
    Lcd_Out(2,1,txt);
  } while(1);
}
```

Precision Internal Oscillator: 8 MHz
Software selectable frequency slow to 31 kHz
Clock Source modes are configured by the FOSC<2:0> bits in the Configuration Word Register 1 (CONFIG1).

REGISTER 14-1: CONFIG1: CONFIGURATION WORD REGISTER 1

—	—	$\overline{\text{DEBUG}}$	LVP	FCMEN	IESO	BOREN1	BOREN0
bit 15							bit 8

$\overline{\text{CPD}}$	$\overline{\text{CP}}$	MCLRE	$\overline{\text{PWRTE}}$	WDTE	FOSC2	FOSC1	FOSC0
bit 7							bit 0

FOSC<2:0>: Oscillator Selection bits

111 =	RC oscillator: CLKOUT function on RA6/OSC2/CLKOUT pin, RC on RA7/OSC1/CLKIN
110 =	RCIO oscillator: I/O function on RA6/OSC2/CLKOUT pin, RC on RA7/OSC1/CLKIN
101 =	INTOSC oscillator: CLKOUT function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN
100 =	INTOSCIO oscillator: I/O function on RA6/OSC2/CLKOUT pin, I/O function on RA7/OSC1/CLKIN
011 =	EC: I/O function on RA6/OSC2/CLKOUT pin, CLKIN on RA7/OSC1/CLKIN
010 =	HS oscillator: High-speed crystal/resonator on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN
001 =	XT oscillator: Crystal/resonator on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN
000 =	LP oscillator: Low-power crystal on RA6/OSC2/CLKOUT and RA7/OSC1/CLKIN

REGISTER 4-1: OSCCON: OSCILLATOR CONTROL REGISTER

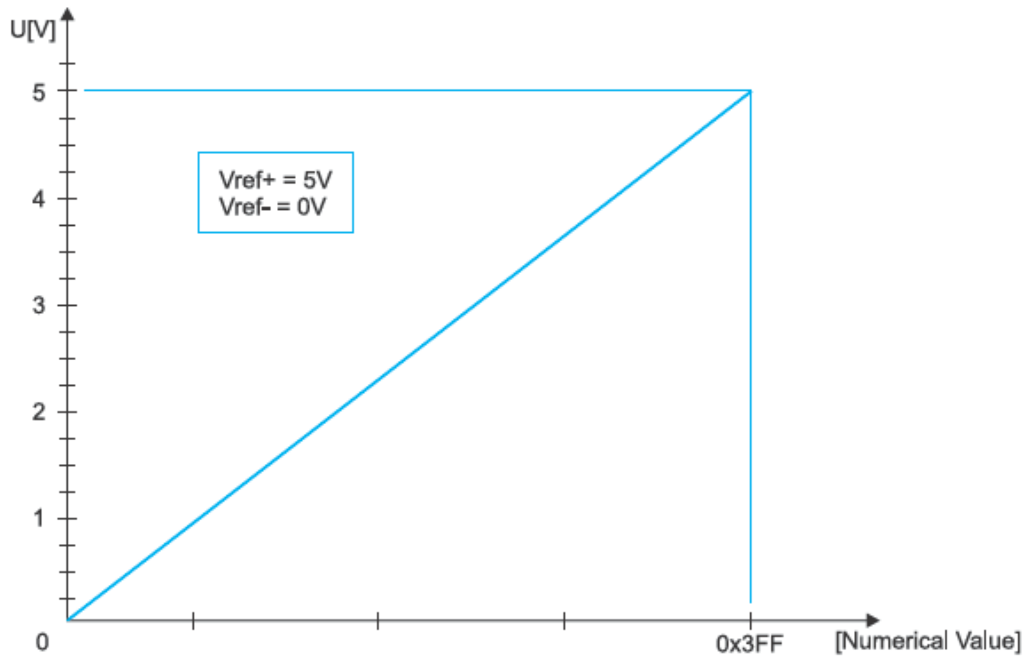
U-0	R/W-1	R/W-1	R/W-0	R-1	R-0	R-0	R/W-0
—	IRCF2	IRCF1	IRCF0	OSTS(1)	HTS	LTS	SCS
bit 7							bit 0

The Oscillator Control (OSCCON) register controls the system clock and frequency selection options. The OSCCON register contains the following bits:

3.9 ANALOG MODULES

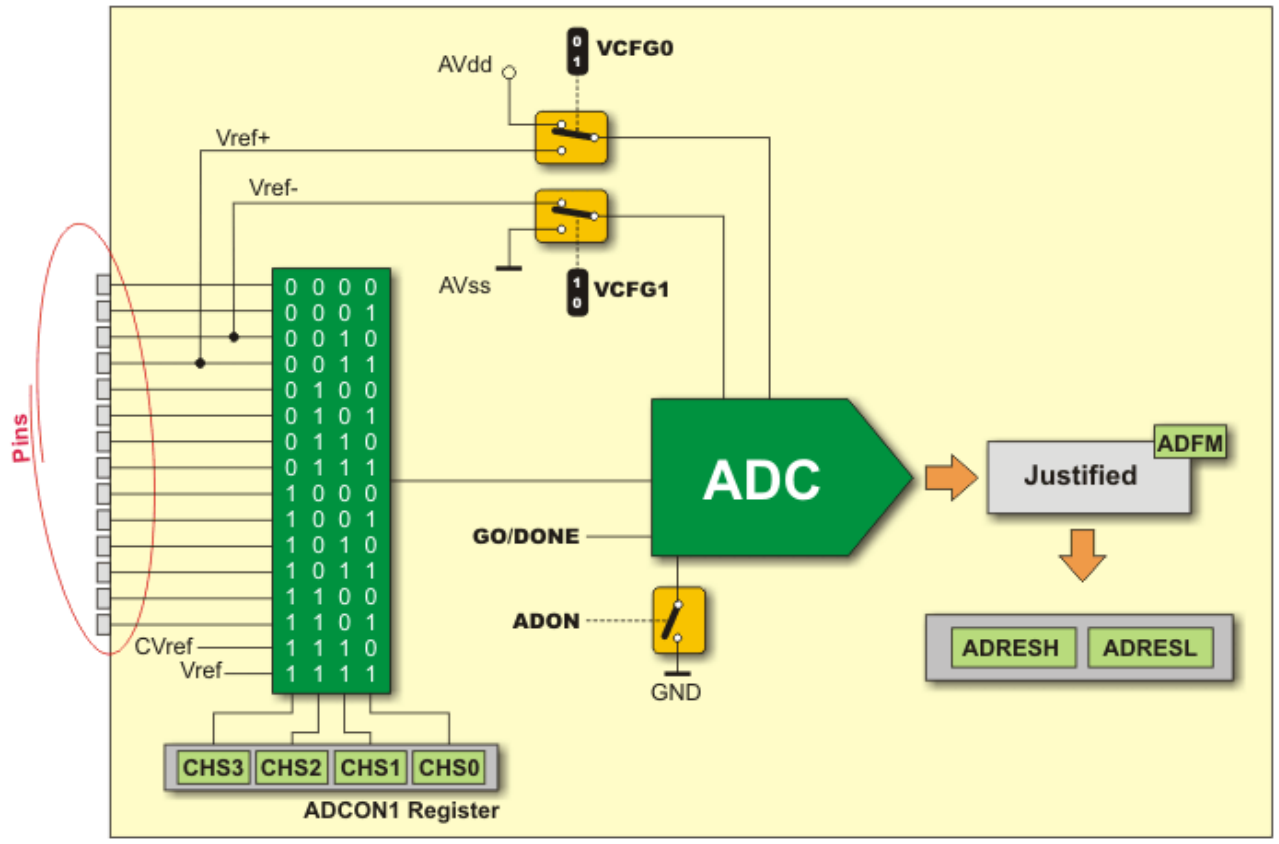
The A/D converter module has the following features:

- The converter generates a 10-bit binary result using the method of successive approximation and stores the conversion results into the ADC registers (ADRESL and ADRESH);
- There are 14 separate analog inputs;
- The A/D converter converts an analog input signal into a 10-bit binary number;
- The minimum resolution or quality of conversion may be adjusted to various needs by selecting voltage references Vref- and Vref+.



A/D CONVERTER

Even though the use of A/D converter seems to be very complicated, it is basically very simple, simpler than using timers and serial communication module, anyway.

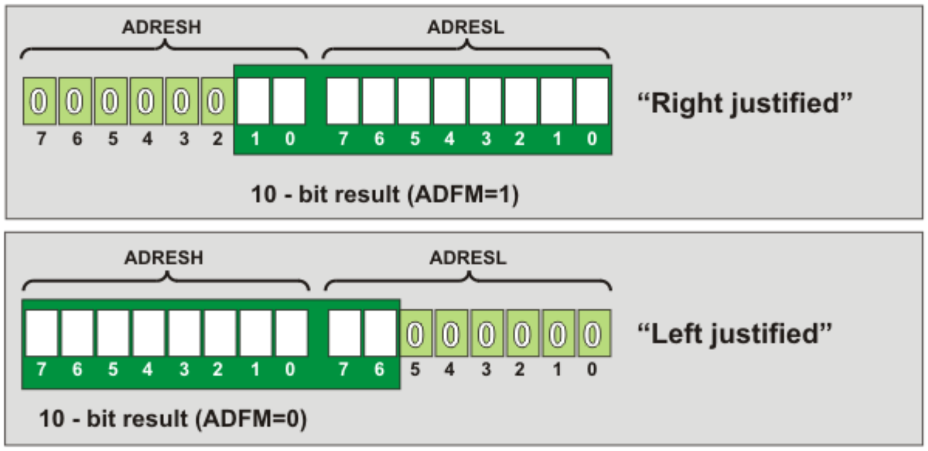


The operation of A/D converter is in control of the bits of four registers:

- ADRESH Contains high byte of conversion result;
- ADRESL Contains low byte of conversion result;
- ADCON0 Control register 0; and
- ADCON1 Control register 1.

ADRESH and ADRESL Registers

The result obtained after converting an analog value into digital is a10-bit number that is to be stored in the ADRESH and ADRESL registers. There are two ways of handling it - left and right justification which simplifies its use to a great extent. The format of conversion result depends on the ADFM bit of the ADCON1 register. In the event that the A/D converter is not used, these registers may be used as general-purpose registers.



A/D ACQUISITION REQUIREMENTS

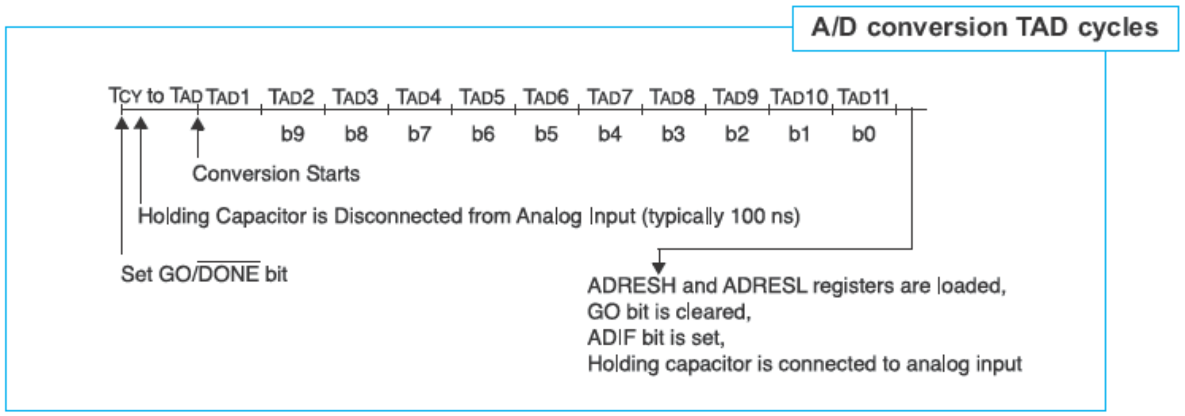
In order to enable the ADC to meet its specified accuracy, it is necessary to provide a certain time delay between selecting specific analog input and measurement itself. This time is called 'acquisition time' and mainly depends on the source impedance. There is an equation used to calculate this time accurately, which in the worst case amounts to approximately 20uS. So, if you want the conversion to be accurate, don't forget this important detail.

ADC CLOCK PERIOD

The time needed to complete a one-bit conversion is defined as TAD. It is required to be at least 1,6 uS. One full 10-bit A/D conversion is slightly longer than expected and amounts to 11 TAD periods. Since both clock frequency and source of A/D conversion are specified by software, it is necessary to select one of the available combinations of bits ADCS1 and ADCS0 before the voltage measurement on some of the analog inputs starts. These bits are stored in the ADCON0 register.

ADC Clock Source	**ADCS1**	**ADCS0**	**Device Frequency (Fosc)**			
			20 Mhz	**8 Mhz**	**4 Mhz**	**1 Mhz**
Fosc/2	0	0	100 nS	250 nS	500 nS	2 uS
Fosc/8	0	1	400 nS	1 uS	2 uS	8 uS
Fosc/32	1	0	1.6 uS	4 uS	8 uS	32 uS
Frc	1	1	2 - 6 uS	2 - 6 uS	2 - 6 uS	2 - 6 uS

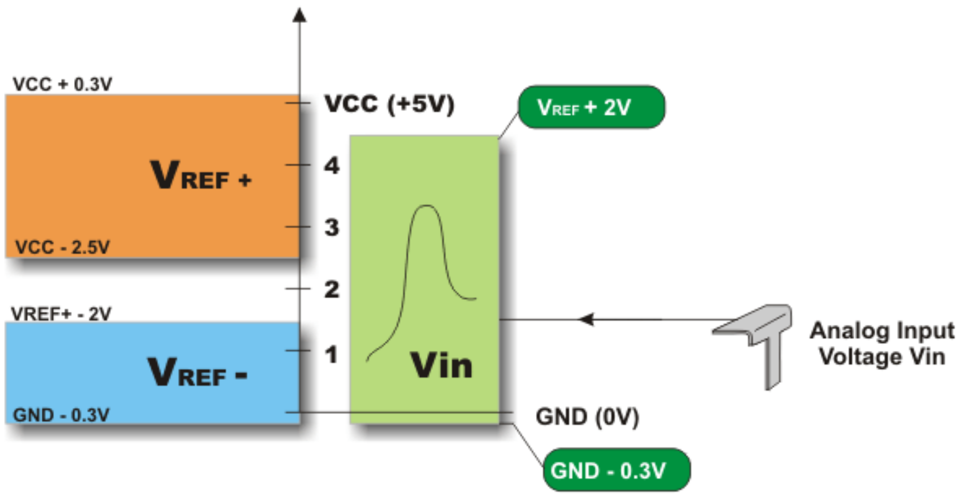
Any change in the system clock frequency will affect the ADC clock frequency, which may adversely affect the ADC result. Device frequency characteristics are shown in the table above. The values in the shaded cells are outside of the range recommended.


A/D conversion TAD cycles
TCY to TAD
TAD1
TAD2
TAD3
TAD4
TAD5
TAD6
TAD7
TAD8
TAD9
TAD10
TAD11
b9
b8
b7
b6
b5
b4
b3
b2
b1
b0
Conversion Starts
Holding Capacitor is Disconnected from Analog Input (typically 100 ns)
Set GO/DONE bit
ADRESH and ADRESL registers are loaded,
GO bit is cleared,
ADIF bit is set,
Holding capacitor is connected to analog input

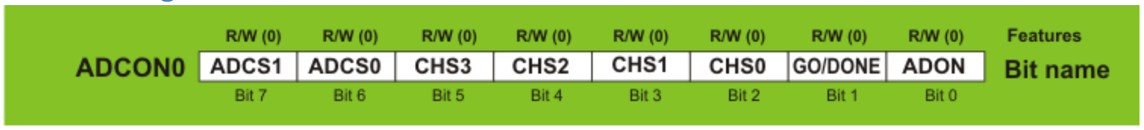
HOW TO USE THE A/D CONVERTER?

In order to enable the A/D converter to run without problems as well as to avoid unexpected results, it is necessary to consider the following:

- A/D converter does not differ between digital and analog signals. In order to avoid errors in measurement or chip damage, pins should be configured as analog inputs before the process of conversion starts. Bits used for this purpose are stored in the TRIS and ANSEL (ANSELH) registers;
- When reading the port with analog inputs, the state of the corresponding bits will be read as a logic zero (0); and
- Roughly speaking, voltage measurement in the converter is based on comparing input voltage with internal scale which has 1024 marks (2^{10} = 1024). The lowest scale mark stands for the Vref- voltage, whilst its highest mark stands for the Vref+ voltage. Figure below shows selectable voltage references as well as their minimum and maximum values.



ADCON0 Register



	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	R/W (0)	Features
ADCON0	ADCS1	ADCS0	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON	Bit name
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	

Legend

R/W	Readable/Writable bit
(0)	After reset, bit is cleared

ADCS1, ADCS0 - A/D Conversion Clock Select bits select clock frequency used for internal synchronization of A/D converter. It also affects duration of conversion.

ADCS1	ADCS2	Clock
0	0	Fosc/2
0	1	Fosc/8
1	0	Fosc/32
1	1	RC *

* Clock is generated by internal oscillator which is built in the converter.

CHS3-CHS0 - Analog Channel Select bits select a pin or an analog channel for A/D conversion, i.e. voltage measurement:

CHS3	CHS2	CHS1	CHS0	Channel	Pin
0	0	0	0	0	RA0/AN0
0	0	0	1	1	RA1/AN1
0	0	1	0	2	RA2/AN2
0	0	1	1	3	RA3/AN3
0	1	0	0	4	RA5/AN4
0	1	0	1	5	RE0/AN5
0	1	1	0	6	RE1/AN6
0	1	1	1	7	RE2/AN7
1	0	0	0	8	RB2/AN8
1	0	0	1	9	RB3/AN9
1	0	1	0	10	RB1/AN10
1	0	1	1	11	RB4/AN11
1	1	0	0	12	RB0/AN12
1	1	0	1	13	RB5/AN13

1	1	1	0	CVref
1	1	1	1	Vref = 0.6V

GO/DONE - A/D Conversion Status bit determines current status of conversion:

- 1 - A/D conversion is in progress.
- 0 - A/D conversion is complete. This bit is automatically cleared by hardware when the A/D conversion is complete.

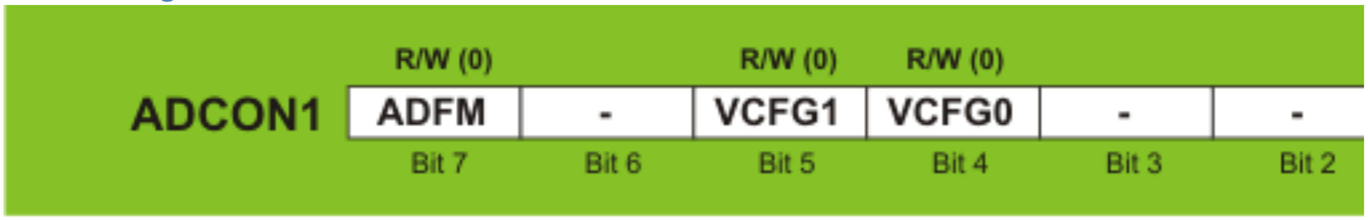
ADON - A/D On bit enables A/D converter.

- 1 - A/D converter is enabled.
- 0 - A/D converter is disabled.

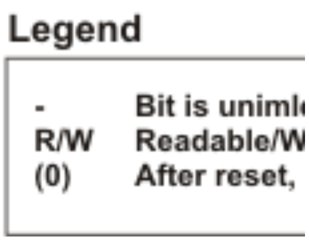
Let's do it in mikroC...

```
/* This example code reads analog value from channel 2 and displays it
on PORTB and
PORTC as 10-bit binary number.*/

#include <built_in.h>
unsigned int adc_rd;

void main() {
    ANSEL = 0x04;               // Configure AN2 as analog pin
    TRISA = 0xFF;               // PORTA is configured as input
    ANSELH = 0;                 // Configure all other AN pins as
digital I/O
    TRISC = 0x3F;               // Pins RC7 and RC6 are configured as
outputs
    TRISB = 0;                  // PORTB is configured as an output

    do {
        temp_res = ADC_Read(2); // Get 10-bit result of AD conversion
        PORTB = temp_res;       // Send lower 8 bits to PORTB
        PORTC = temp_res >> 2;  // Send 2 most sig. bits to RC7,RC6
    } while(1);                 // Remain in the loop
}
```

ADCON1 Register



ADCON1	R/W (0)		R/W (0)	R/W (0)		
	ADFM	-	VCFG1	VCFG0	-	-
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2



Legend

-	Bit is uniml
R/W	Readable/W
(0)	After reset,

ADFM - A/D Result Format Select bit

- 1 - Conversion result is right justified. Six most significant bits of the ADRESH are not used.
- 0 - Conversion result is left justified. Six least significant bits of the ADRESL are not used.

VCFG1 - Voltage Reference bit selects negative voltage reference source needed for the operation of A/D converter.

- 1 - Negative voltage reference is applied to the Vref- pin.
- 0 - Power supply voltage Vss is used as negative voltage reference source.

VCFG0 - Voltage Reference bit selects positive voltage reference source needed for the operation of A/D converter.

- 1 - Positive voltage reference is applied to the Vref+ pin.
- 0 - Power supply voltage Vdd is used as positive voltage reference source.

In Short

In order to measure voltage on an input pin by the A/D converter, the following should be done:

Step 1 - Port configuration:

- Write a logic one (1) to a bit of the TRIS register, thus configuring the appropriate pin as an input.
- Write a logic one (1) to a bit of the ANSEL register, thus configuring the appropriate pin as an analog input.

Step 2 - ADC module configuration:

- Configure voltage reference in the ADCON1 register.
- Select ADC conversion clock in the ADCON0 register.
- Select one of input channels CH0-CH13 of the ADCON0 register.
- Select data format using the ADFM bit of the ADCON1 register.
- Enable A/D converter by setting the ADON bit of the ADCON0 register.

Step 3 - ADC interrupt configuration (optionally):

- Clear the ADIF bit.
- Set the ADIE, PEIE and GIE bits.

Step 4 - Wait for the required acquisition time to pass (approximately 20uS).

Step 5 - Start conversion by setting the GO/DONE bit of the ADCON0 register.

Step 6 - Wait for ADC conversion to complete.

- It is necessary to check in the program loop whether the GO/DONE pin is cleared or wait for an A/D interrupt (must be previously enabled).

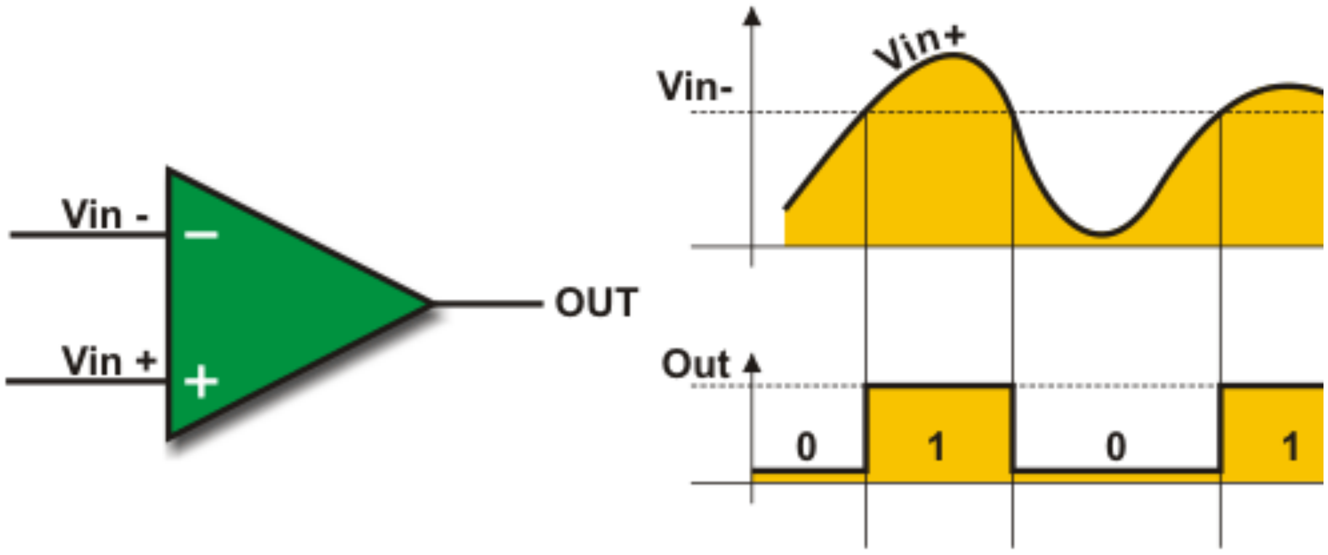
Step 7 - Read ADC results:

- Read the ADRESH and ADRESL registers.

ANALOG COMPARATOR

In addition to A/D converter, there is another module, which until quite recently has been embedded only in integrated circuits belonging to the so called analog electronics. Owing to the fact that it is hardly possible to find any more complex automatic device which in some way does not use these circuits, two high quality comparators, along with additional electronics, are integrated into the microcontroller and connected to its pins.

How does a comparator operate? Basically, the analog comparator is an amplifier which compares the magnitude of voltages at two inputs. It has two inputs and one output. Depending on which input has a higher voltage (analog value), a logic zero (0) or logic one (1) (digital values) will appear on its output:



- When the analog voltage at Vin- is higher than that at Vin+, the output of the comparator is a digital low level.
- When the analog voltage at Vin+ is higher than that at Vin-, the output of the comparator is a digital high level.

The PIC16F887 microcontroller has two such voltage comparators the inputs of which are connected to I/O pins RA0-RA3, whereas the outputs are connected to the RA4 and RA5 pins. There is also a voltage reference internal source on the chip itself, which will be discussed later.

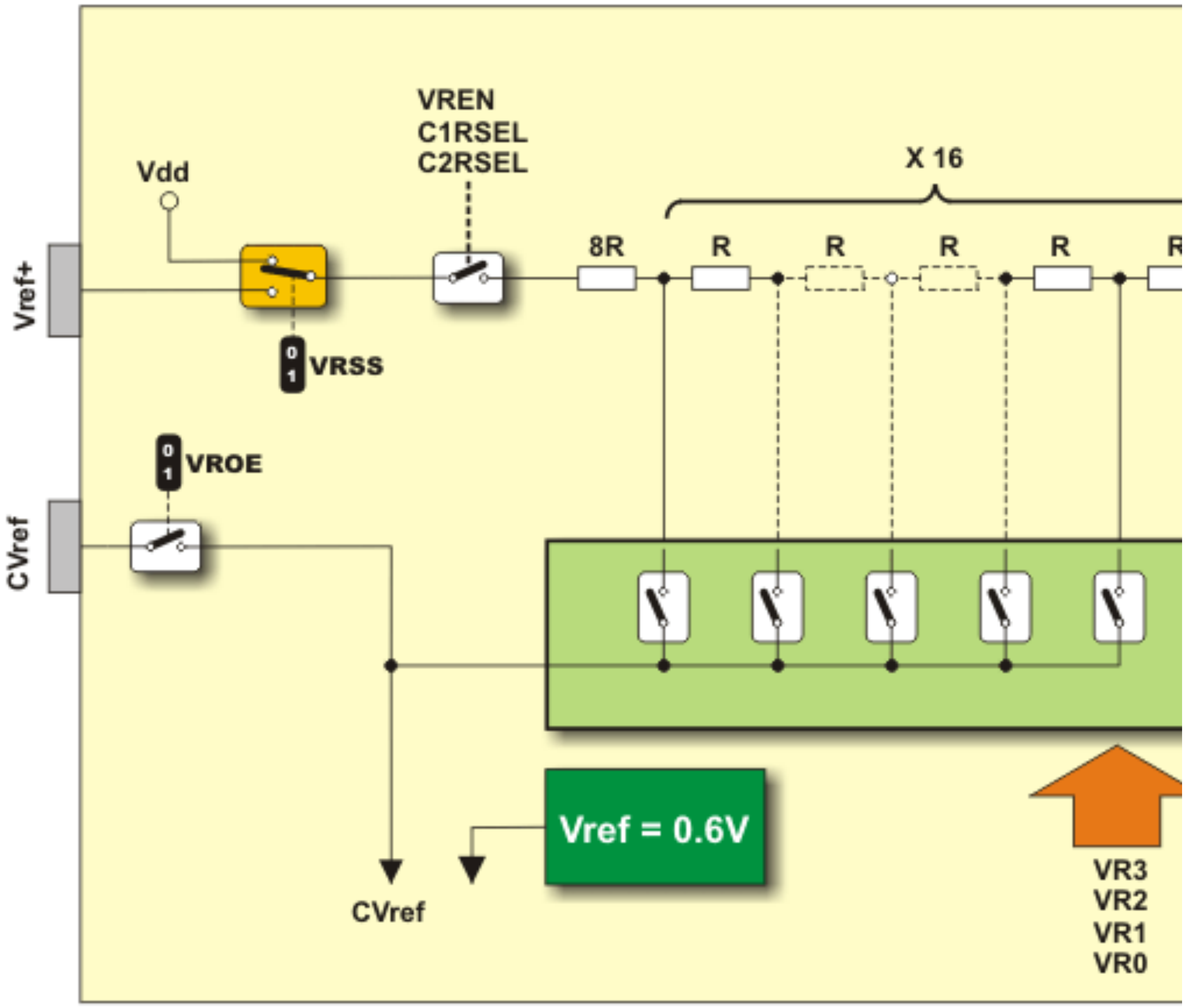
These two circuits are under control of the bits stored in the following registers:

- CM1CON0 is in control of comparator C1;
- CM2CON0 is in control of comparator C2;
- CM2CON1 is in control of comparator C2;

VOLTAGE REFERENCE INTERNAL SOURCE

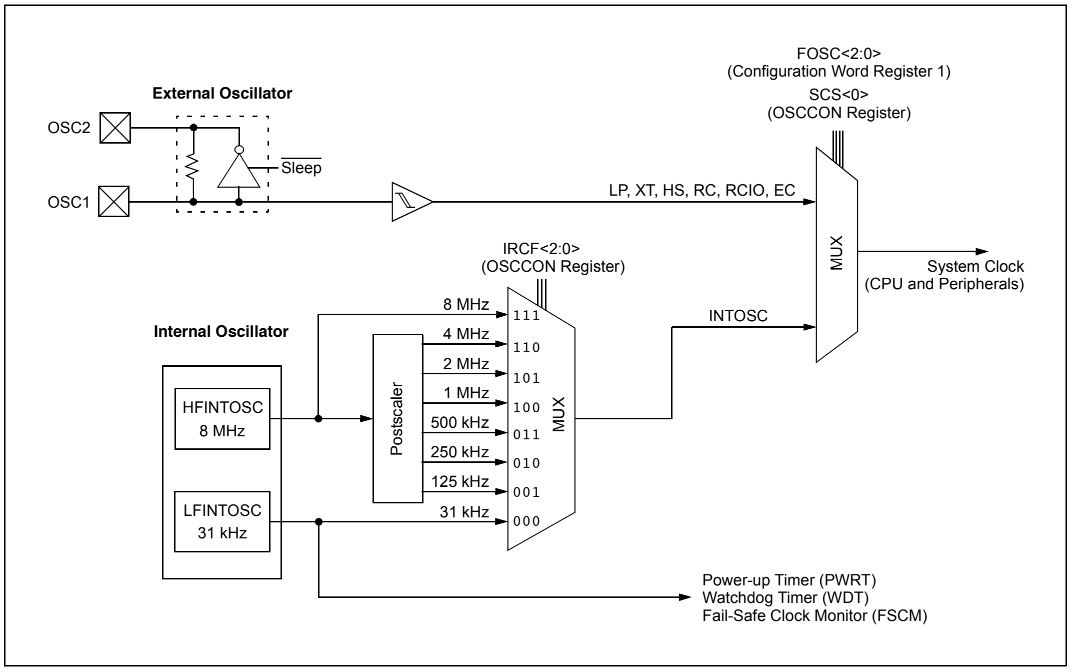
One of two analog voltages provided on the comparator inputs is usually stable and unchangeable. It is called '*voltage reference*'(Vref). To generate it, both external and special internal voltage source can be used. When the voltage source is selected, Vref is derived from it by means of a ladder network consisting of 16 resistors which form a voltage divider. The voltage source is selectable through the both ends of the divider by the VRSS bit of the VRCON register.

In addition, the voltage fraction provided by the resistor ladder network may be selected through the bits VR0-VR3 and used as a voltage reference. See figure below.


VREN
C1RSEL
C2RSEL
Vdd
X 16
Vref+
8R
R
R
R
R
R
VRSS
VROE
CVref
Vref = 0.6V
CVref
VR3
VR2
VR1
VR0

The compara

FIGURE 4-1: SIMPLIFIED PIC® MCU CLOCK SOURCE BLOCK DIAGRAM



8MHz clock is about 125 ns the recommended TAD is about 1.6 usec or 10 times the 8MHZ