Chapter 4 - ADC – "take-aways"

- 2 steps in ADC: Acquire (not too long, not too short) + Convert

- Converting from analogue input to digital output

  Formula: Digital result = [Vin – $Vref_-$] / [$Vref_+$ - $Vref_-$] x $[2^{10} - 1]$

  If $Vref_+$ = 5 V, $Vref_-$ = 0 V, Digital result = Vin / 5 x 1023

- **ADCON0**

-	-	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON

Select the analogue input channel to convert. E.g. CHS3..CHS0 = **0000** means convert AN0.

Set GO = 1 to start the conversion.

Check GO == 0 for completion of conversion.

Set ADON = **1** to turn on the ADC module.

- **ADCON1**

-	-	VCFG1	VCFG0	PCFG3	PCFG2	PCFG1	PCFG0

Set VCFG1 = **0**, VCFG0 = **0** to use 0 volt & 5 volts as reference voltages.

Partition the I/O pins into digital/analogue use. E.g. PCFG3..PCFG0 = **1110** means use AN0 as analogue input, the rest are digital.

- **ADCON2**

ADFM	-	ACQT2	ACQT1	ACQT0	ADCS2	ADCS1	ADCS0

= 1 means right justified.

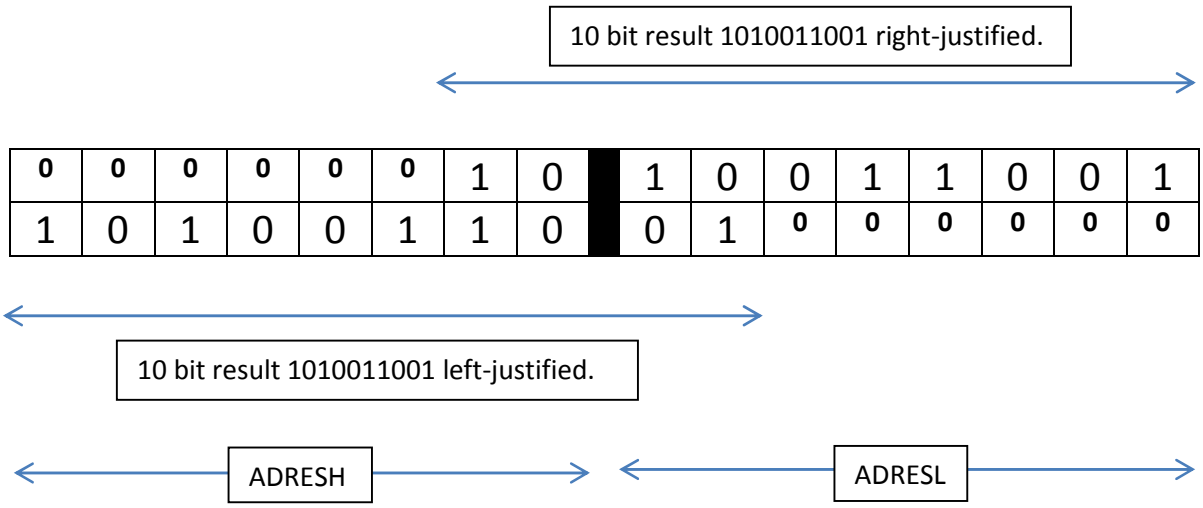
= **0** means left-justified.

Set the acquisition time. E.g. ACQT2..ACQT0 =**010** means use 4 T_{AD}. = 4 x 1.33 = 5.3 us.

(Minimum acquisition time depends on a number of factors. Sample calculation in PIC18F4550 datasheet gives 2.45 us as minimum.)

Select the ADC clock source. E.g. ADCS2..ADCS0 =**110** means use Fosc/64. So T_{AD} = 64/Fosc = 64/48 = 1.33 us. (min is 0.7 us.)

- **ADRESH** & **ADRESL**



- Last page of C4 notes → a **typical program**.

```
main (void)
{
  TRISD = 0x00;        // configure Port D as output - for LED bar
  PORTD = 0x00;        // initialise Port D – all LED's off
  // configure A/D converter module & switch it on
  ADCON0 = 0 b 0 0 0 0 0 0 1;     // bits <5:2> = 0000, channel AN0 selected
                                  // bit <0> = 1, A/D activated (powered up)
  ADCON1 = 0 b 0 0 0 0 1 1 1 0;   // bit <5> = 0, Vref- = Vss (0V)
                                  // bit <4> = 0, Vref+ = Vdd (5V)
                                  // bits <3:0> = 1110, pin AN0 as analogue input
  ADCON2 = 0 b 0 0 0 1 0 1 1 0;   // bit <7> = 0, result left justified
                                  // bits <5:3> = 010, acquisition time = 4 TAD
                                  // bits <2:0> =  110, conversion clock = FOSC/64
  while (1)
    {
    ADCON0bits.GO = 1;            // start A/D conversion
    while (ADCONbits.GO == 1);    // wait here for /DONE to becomes 0
    PORTD = ADRESH;               // output upper 8-bit of result to LED bar
    }
```