Tahmid

www.edaboard.com/nextnewesttoentry1570.html

Right/Left Justification continued

When you have a 10-bit number (range is 0 to 1023), if you divide the number by 4, you effectively just converted the 10-bit number to an 8-bit number. The new range is 0 to 255.

You should already know that bit-shifting a number to the right once is equivalent to dividing the number by 2 and bit-shifting the number to the right twice is equivalent to dividing the number by 4. If you don't know about this, this should help you understand.

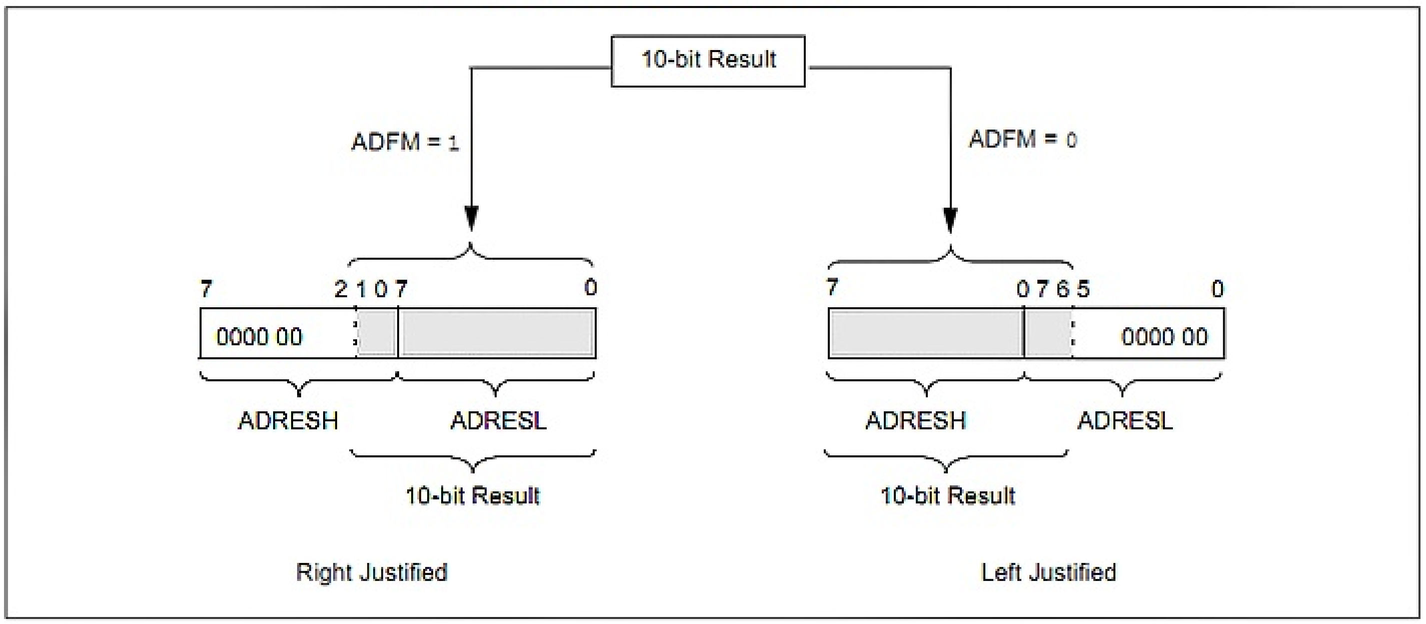
Let's take the number 128. In binary, it is represented as 1000 0000. If we shift it to the right twice, we have 10 0000, which is the same as 0010 0000. 0010 0000 is equivalent to 32 in decimal. 128/32 = 4. So, by bit-shifting 128 to the right twice, we divided it by 4.

Let's take a more complicated number: 255. In binary it is represented as 1111 1111. If we shift it to the right twice, we have 0011 1111. This is equivalent to 63. We now know that bit-shifting to the right twice means dividing by 4. 255/4 = 63.75. For the PIC, with simple integer math, this is stored as 63. See, we get the same value when dividing by 4 or bit-shifting to the right twice.

So, if we have a 10-bit number and shift it to the right twice, we get an 8-bit number. We divided the number by four and got rid of the lowest two bits as these get lost as they are shifted out. What we have left is the 8-bit number which is just the highest 8 bits of the 10 bits with the lowest two bits discarded.

Now think about ADRESH and ADRESL when the ADC conversion result is stored left-justified. So, ADRESH stores the higher 8 bits and ADRESL stores the lower 2 bits along with 6 zeroes. If we do not consider ADRESL and think of only ADRESH, it contains the higher 8 bits of the 10 bit ADC conversion result. So, if that 10 bit result is divided by 4 or bit-shifted twice to the right to an 8-bit number, the result is the upper 8 bits, which is what is stored in ADRESH. So, if we do not consider ADRESL and consider only ADRESH, ADRESH stores the 8-bit equivalent value of the 10-bit ADC result. An example should clear this.

A/D RESULT JUSTIFICATION



Example:

A circuit for analogue to digital conversion is set up, to utilize the internal ADC of the PIC16F877A. VREF+ = 5.0V, VREF- = 0V. The value of TAD is selected such that TAD > 1.6µs. An input voltage of 2.0V is measured by the ADC and the corresponding value is stored in the registers ADRESH and ADRESL. What are the values of ADRESH and ADRESL, if the result is left-justified ?

Solution:
ADC conversion result = (2.0/5.0)*1023 = 409
ADRESH stores the upper 8 bits. 409 in binary is 01 1001 1001. The lower 2 bits are stored in ADRESL So, ADRESL stores 0100 0000. ADRESH stores 0110 0110.

ADRESH stores 0110 0110, which is equivalent to the decimal number 102. The 10-bit result was 409. 409/4 = 102 for the PIC. This is what ADRESH stores. If we reassess the situation, assuming, for the time being, that the PIC ADC has a resolution of 8-bits and not 10-bits. So, the conversion result is in the range 0 to 255, 255 being the maximum possible value.

So, ADC conversion result for the above problem = (2.0/5.0)*255 = 102

This is what ADRESH stores. So, ADRESH stores the 8-bit equivalent of the 10-bit ADC conversion result.

Continued in Part 4.............