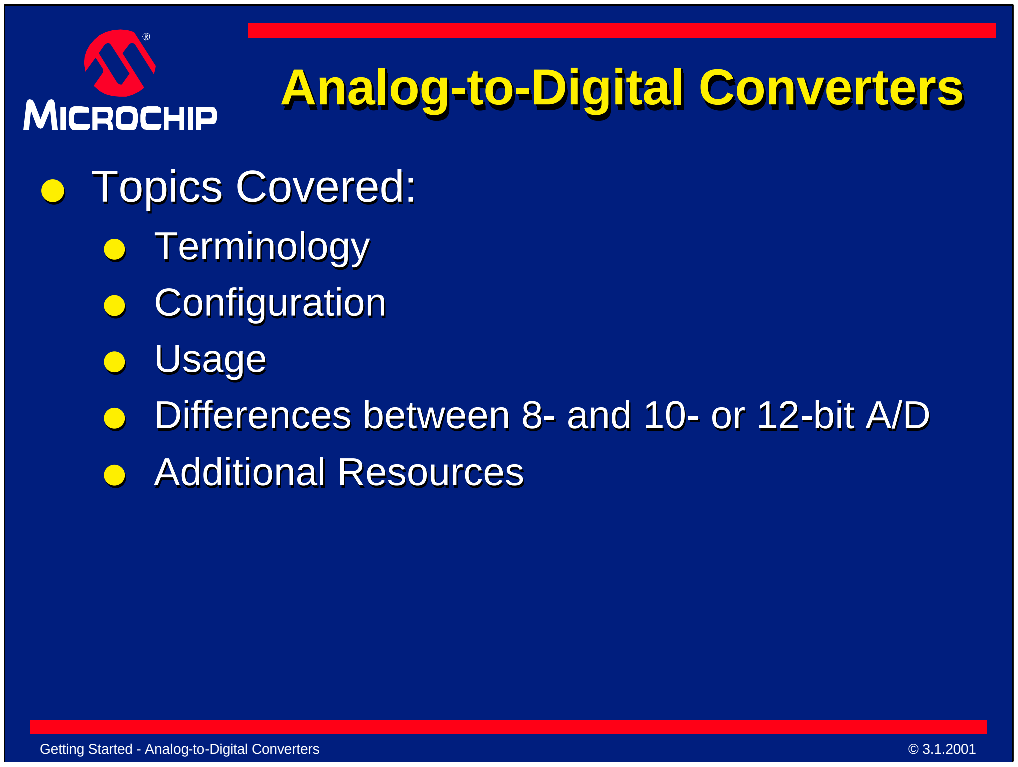
Analog-to-Digital Converters

In this presentation we will look at the Analog-to-Digital Converter Peripherals with Microchip’s midrange PICmicro® Microcontrollers series.

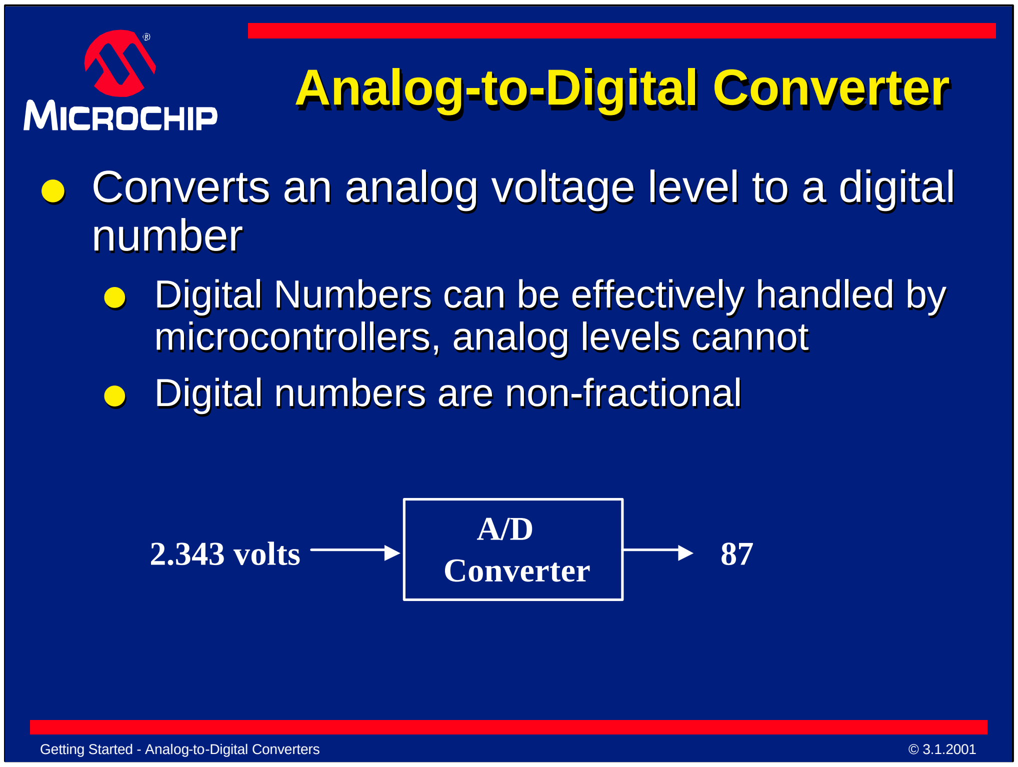


Analog-to-Digital Converters

- Topics Covered:
  - Terminology
  - Configuration
  - Usage
  - Differences between 8- and 10- or 12-bit A/D
  - Additional Resources

Getting Started - Analog-to-Digital Converters © 3.1.2001

Basic analog-to-digital converter terminology will be covered first, followed by configuration of the analog-to-digital converter peripheral. Next, information on the usage of the peripheral will be presented, initially focusing on the 8-bit analog-to-digital converter. Then the differences between the 8-bit and the 10-or 12-bit converters will be discussed. Finally, some additional reference resources will be highlighted.



Analog-to-Digital Converter

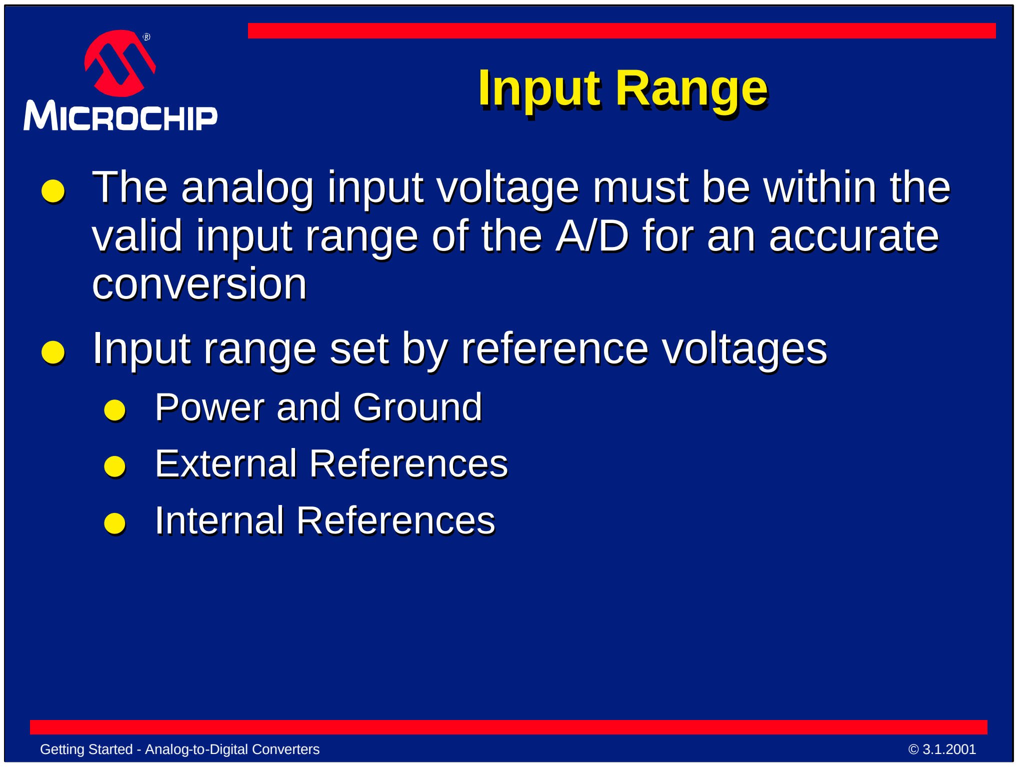
- Converts an analog voltage level to a digital number
  - Digital Numbers can be effectively handled by microcontrollers, analog levels cannot
  - Digital numbers are non-fractional

$$\text{2.343 volts} \rightarrow \boxed{\text{A/D Converter}} \rightarrow 87$$

Getting Started - Analog-to-Digital Converters © 3.1.2001

Microcontrollers are very efficient at processing digital numbers, but they cannot handle analog signals directly. An analog-to-digital converter, converts an analog voltage level to a digital number. The microcontroller can then efficiently process the digital representation of the original analog voltage. By definition, digital numbers are non-fractional whole numbers.

In this example, an input voltage of 2.343 volts is converted to 87. The user's software can use the value 87 as the representation of the original input voltage. At this point, the number 87 is only used for discussion purposes as a typical output.



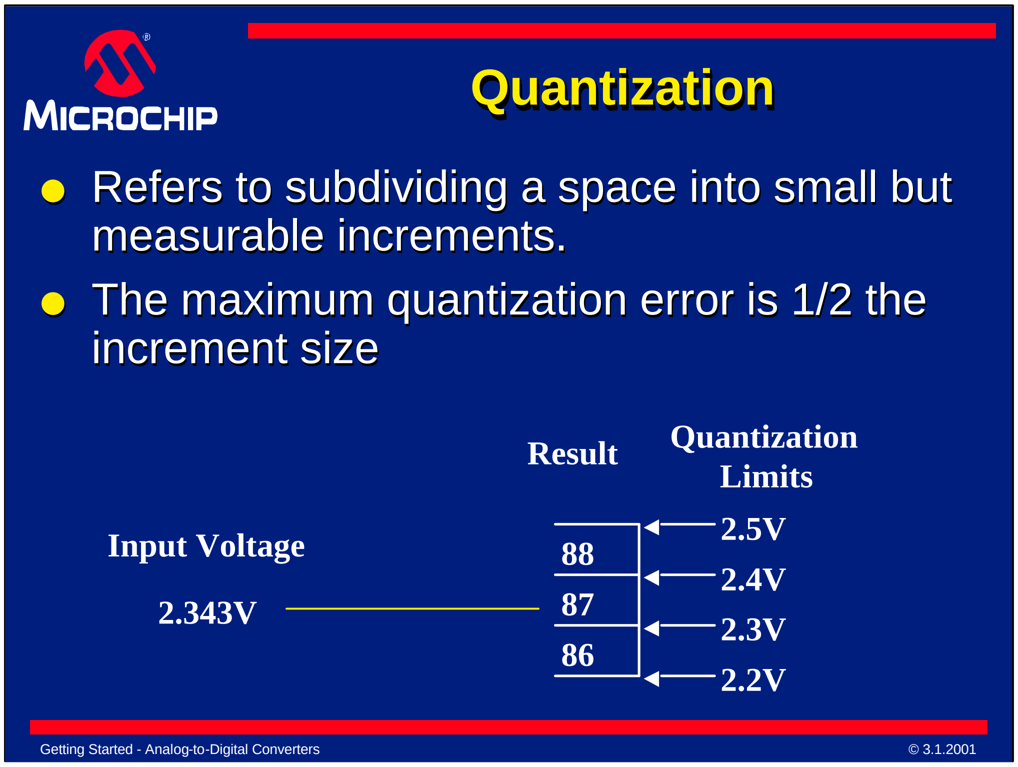
Input Range

- The analog input voltage must be within the valid input range of the A/D for an accurate conversion
- Input range set by reference voltages
  - Power and Ground
  - External References
  - Internal References

Getting Started - Analog-to-Digital Converters © 3.1.2001

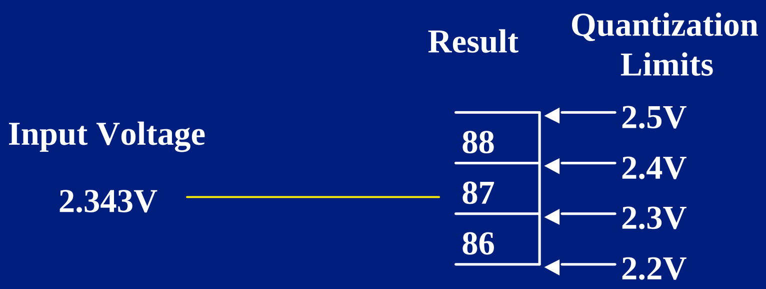
The analog-to-digital converter is only capable of performing an accurate conversion if the analog input voltage is within the valid input range of the converter. If the input voltage falls outside this range, the conversion value will be inaccurate. The input range is set by high and low voltage references. These define the upper and lower limits of the valid input range. In many cases, the high and low voltage references are selected as the microcontroller supply voltage and ground, at other times an external reference or references are used.

In addition, some devices have internal voltage references that can be used. The source or sources for these voltage references are a configuration option when setting up the analog-to-digital converter in the PICmicro microcontroller (MCU). Note that there are restrictions on the voltage reference levels, for example: the reference voltages generally shouldn't be less than Vss or greater than VDD. There is also a minimum difference that is required between the high and low reference voltages. Please consult your data sheet for the voltage reference requirements.



Quantization

- Refers to subdividing a space into small but measurable increments.
- The maximum quantization error is 1/2 the increment size



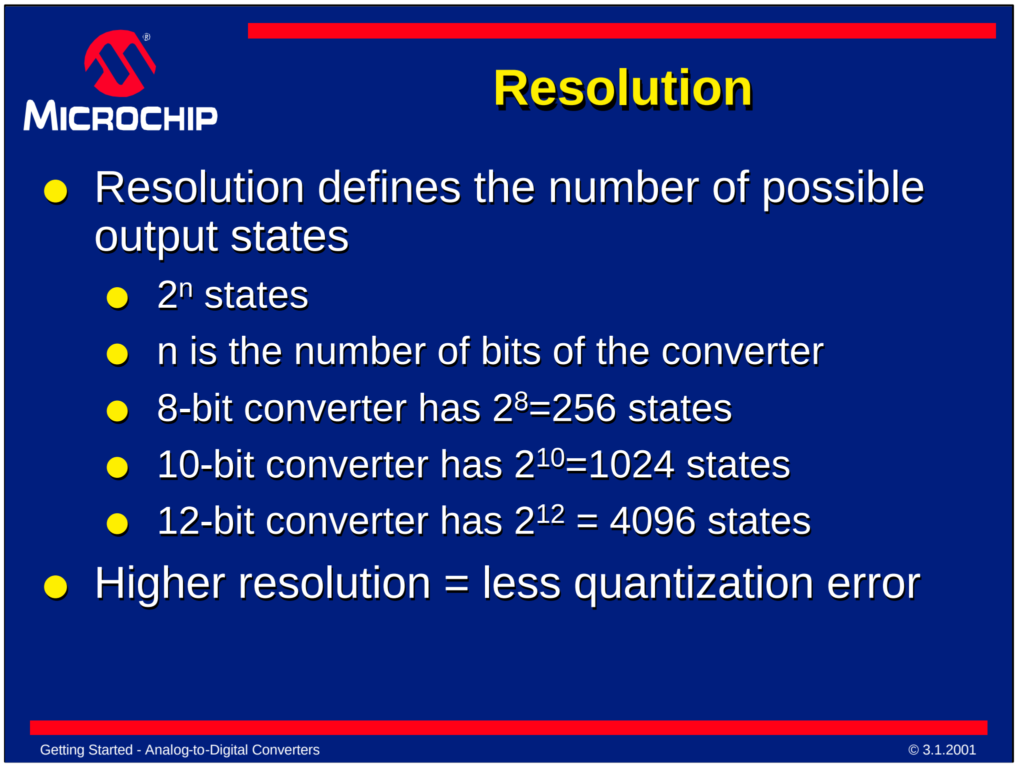
Getting Started - Analog-to-Digital Converters

© 3.1.2001

The output of an analog-to-digital converter is a quantized representation of the original analog signal. The term quantization refers to subdividing a range into small but measurable increments. The total allowable input range is divided into a finite number of regions with a fixed increment. The analog-to-digital converter determines the appropriate region to assign the given input voltage.

In this example, the step or increment is one-tenth of a volt and the input voltage is 2.343 volts. The appropriate result would be assigned as a digital value of 87, because 2.343 volts fits between the quantization limits of 2.3 volts and 2.4 volts. Any input voltage between the 2.3 and 2.4 volt quantization limits will be assigned a digital value of 87.

The process of quantization has the potential to introduce an inaccuracy known as quantization error, which can be viewed as being similar to a rounding error. In the above example, the 2.343 volt input is in effect rounded to the nearest tenth of a volt. The maximum quantization error in this case would be five hundredths of a volt or one-half of the increment size. It should be noted that the minimum quantization error for the analog-to-digital converter peripheral in the PICmicro devices is 500 micro volts. Therefore, the smallest step size for each state cannot be less than one milli-volt.

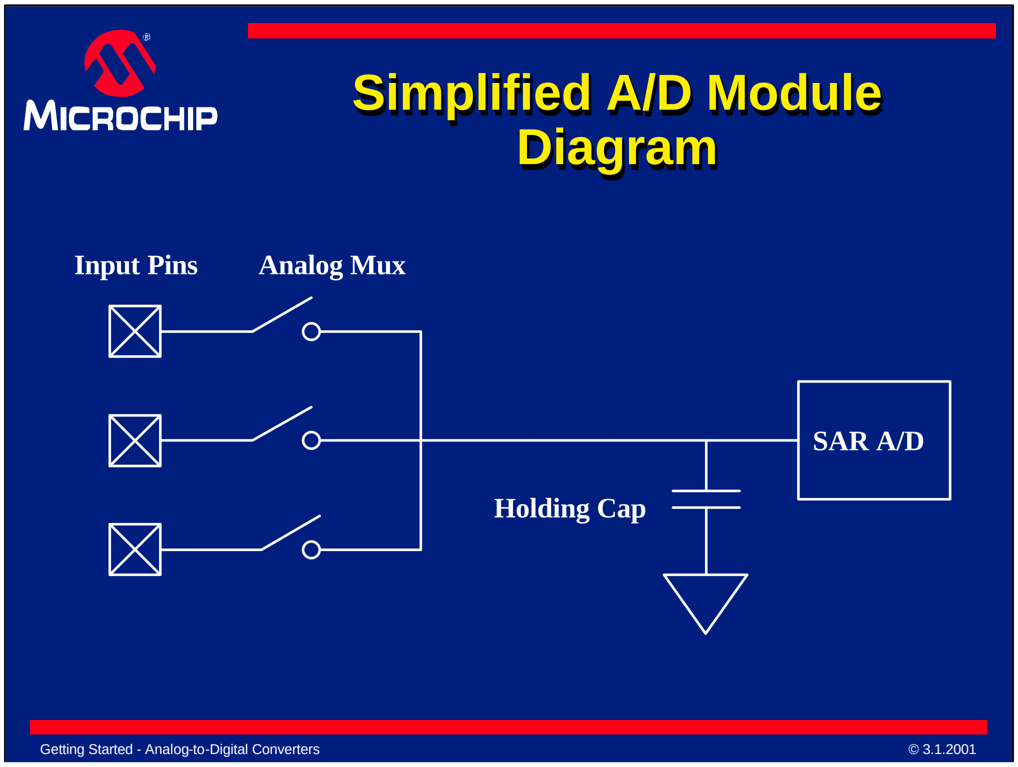


Resolution

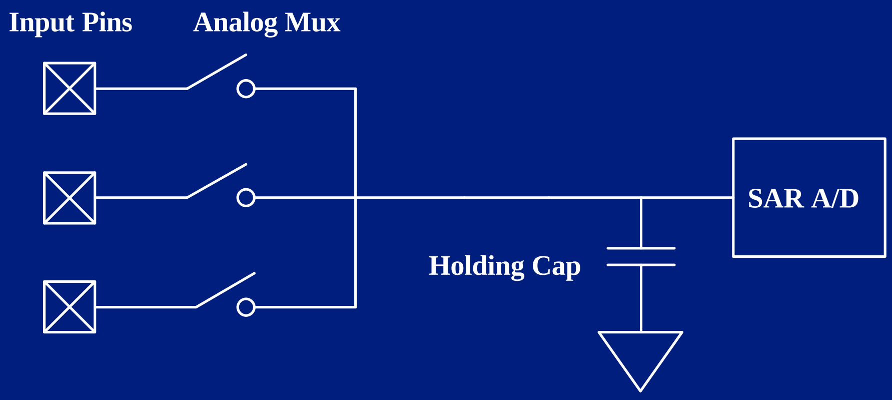
- Resolution defines the number of possible output states
  - 2^n states
  - n is the number of bits of the converter
  - 8-bit converter has 2^8=256 states
  - 10-bit converter has 2^{10}=1024 states
  - 12-bit converter has 2^{12} = 4096 states
- Higher resolution = less quantization error

Getting Started - Analog-to-Digital Converters © 3.1.2001

Resolution defines the number of possible analog-to-digital converter output states. As previously discussed, the result is a digital or whole number, so for an 8-bit converter the possible states will be: zero, one, two, three and so on, with 255 as the maximum state. A 10-bit converter will have 1023 as the maximum state, and a 12-bit converter will have 4095 as the maximum state. If the input range remains constant, a higher resolution converter will have less quantization error because the range is divided into smaller steps. This is similar in concept to the process of rounding a number to the nearest hundredths, having potentially less error than rounding to the nearest tenths.

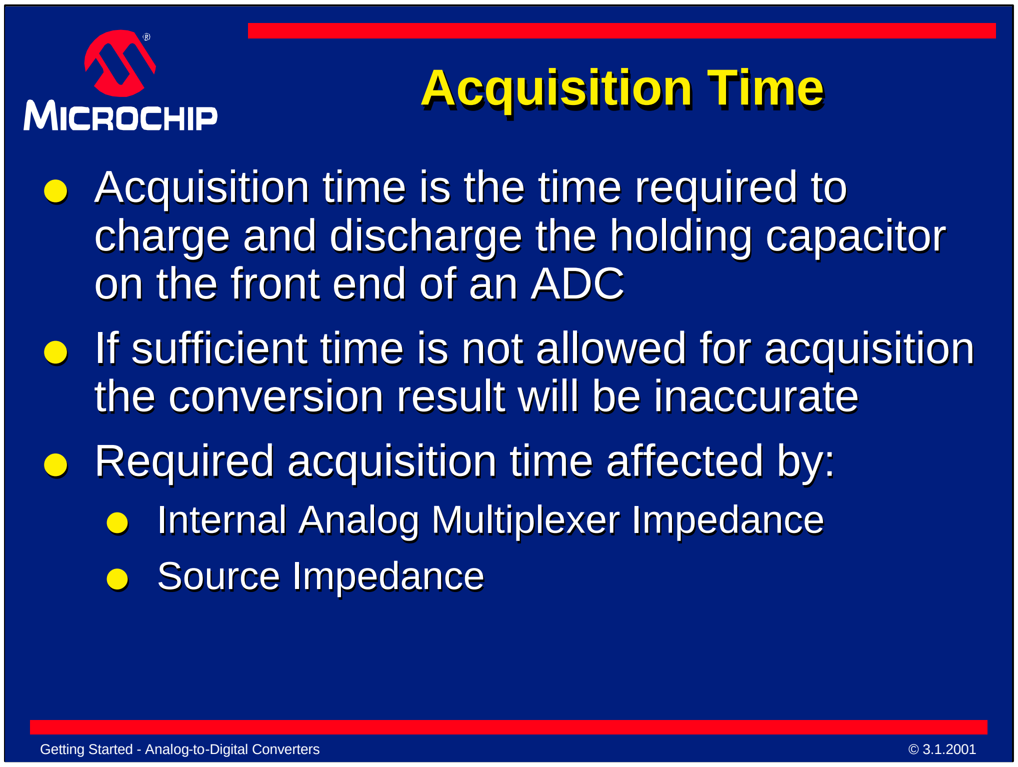


Simplified A/D Module Diagram



This is a simplified diagram of the analog-to-digital converter module. The analog input pins are connected to the inputs of an analog multiplexer which connects the selected channel to the holding capacitor. The analog multiplexer allows multiple inputs to be available for conversion. It is important to note however, that there is only one analog-to-digital converter on the microcontroller, and only one channel can be selected, and therefore converted at a time.

Normally the holding capacitor is connected to the output of the analog multiplexer. When a conversion is initiated, the analog multiplexer disconnects all inputs from the holding capacitor, and the successive approximation converter performs the conversion on the voltage stored on the holding capacitor.

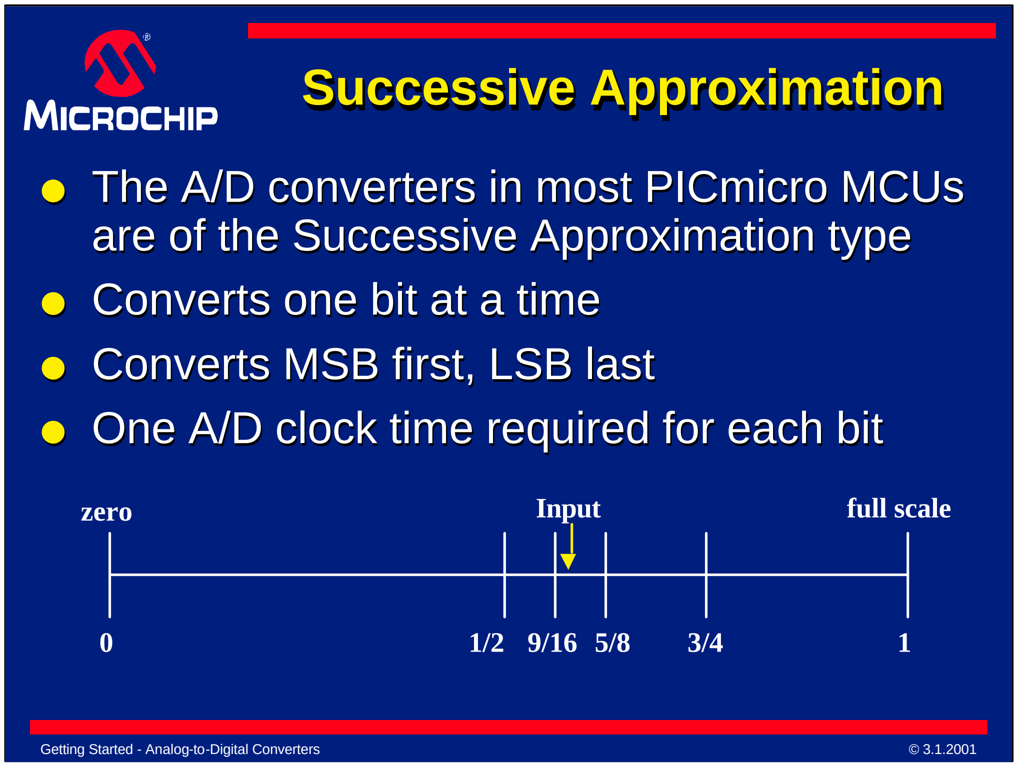


Acquisition Time

- Acquisition time is the time required to charge and discharge the holding capacitor on the front end of an ADC
- If sufficient time is not allowed for acquisition the conversion result will be inaccurate
- Required acquisition time affected by:
  - Internal Analog Multiplexer Impedance
  - Source Impedance

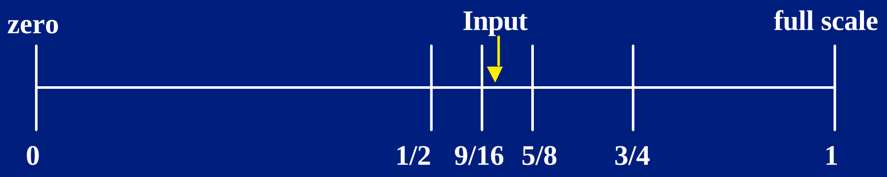
Getting Started - Analog-to-Digital Converters © 3.1.2001

Acquisition time is the amount time required to charge the holding capacitor on the front end of an analog-to-digital converter. The holding capacitor must be given sufficient time to settle to the analog input voltage level before the actual conversion is initiated. If sufficient time is not allowed for acquisition, the conversion will be inaccurate. The required acquisition time is based on a number of factors, two of them being the impedance of the internal analog multiplexer and the output impedance of the analog source. An increase in the source impedance will increase the required acquisition time. In addition, there is a maximum recommended source impedance. This is generally 10K ohms for the 8- and 10-bit converters and 2.5K ohms for the 12-bit devices. Please consult the analog-to-digital converter section of your device data sheet for the equation to calculate the minimum acquisition time for your particular application.



Successive Approximation

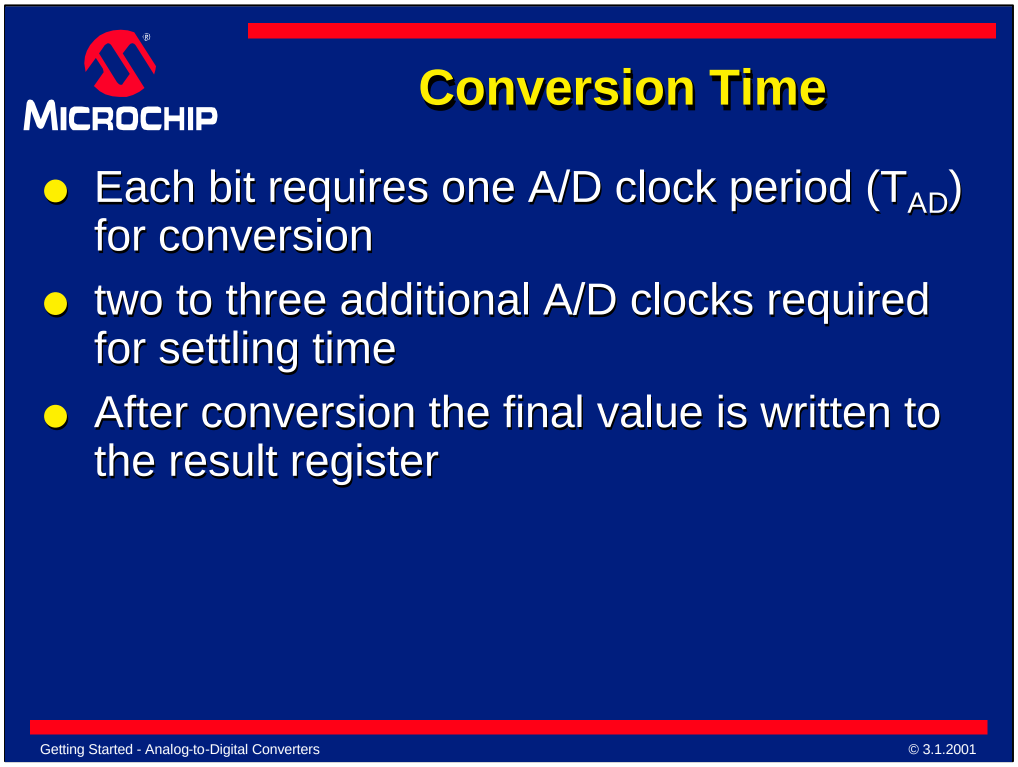
- The A/D converters in most PICmicro MCUs are of the Successive Approximation type
- Converts one bit at a time
- Converts MSB first, LSB last
- One A/D clock time required for each bit



Getting Started - Analog-to-Digital Converters © 3.1.2001

The analog-to-digital converters in most PICmicro MCUs are of the successive approximation type. The successive approximation converter performs a conversion on one bit at a time, beginning with the most significant bit (MSB) and ending with the least significant bit (LSB). The value of the most significant bit is determined by whether the input signal is in the upper or lower half of the valid input range. The next most significant bit is determined by whether the input is in the upper or lower half of the remaining range, and so on, until the least significant bit is determined.

In this example, the input level is indicated by the arrow. For discussion sake, we are using a normalized input range where zero is the minimum valid input level and one is the maximum valid input level. The most significant bit will be converted as a 1, because the input is in the upper half of the range. The next bit would be converted as a 0, because the test range for this bit is between 1/2 and 1, and the input is less than 3/4, which is the mid-point of the test range. The next bit is also converted as a 0, because the test range is between 1/2 and 3/4 and the input is below 5/8 which places it in the lower half of the test range. The following bit will be converted as a 1 because it is above 9/16 placing it in the upper half of the test range. This process is repeated until all the bits are determined. Each bit requires one analog-to-digital converter clock period for conversion.

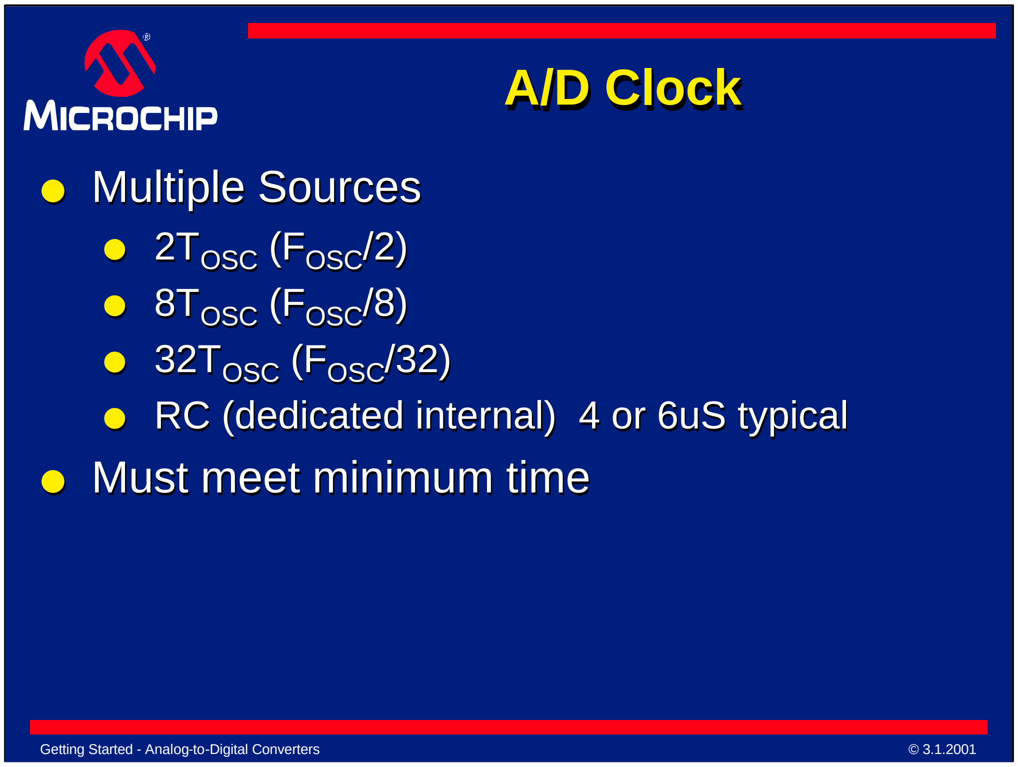


Conversion Time

- Each bit requires one A/D clock period (T_{AD}) for conversion
- two to three additional A/D clocks required for settling time
- After conversion the final value is written to the result register

Getting Started - Analog-to-Digital Converters © 3.1.2001

The time required for the successive approximation conversion and writing of the final value to the result register is the conversion time. This time will be the analog-to-digital clock period multiplied by the number of bits of resolution in the analog-to-digital converter plus the two to three additional clock periods for the settling time. The number of additional clock periods required for the settling time is specified in the analog-to-digital converter section of the specific device data sheet. Upon completion of the conversion, the result is written to the result register for use by the digital circuitry in the microcontroller.

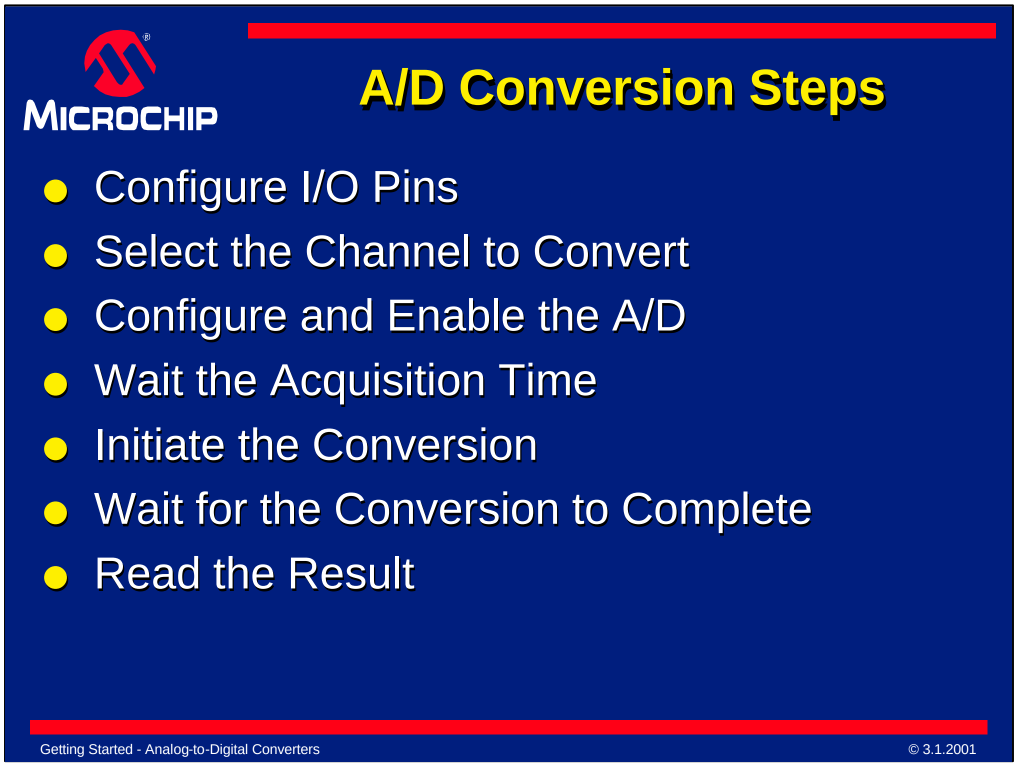


A/D Clock

- Multiple Sources
  - $2T_{OSC}$ ($F_{OSC}/2$)
  - $8T_{OSC}$ ($F_{OSC}/8$)
  - $32T_{OSC}$ ($F_{OSC}/32$)
  - RC (dedicated internal) 4 or 6uS typical
- Must meet minimum time

Getting Started - Analog-to-Digital Converters © 3.1.2001

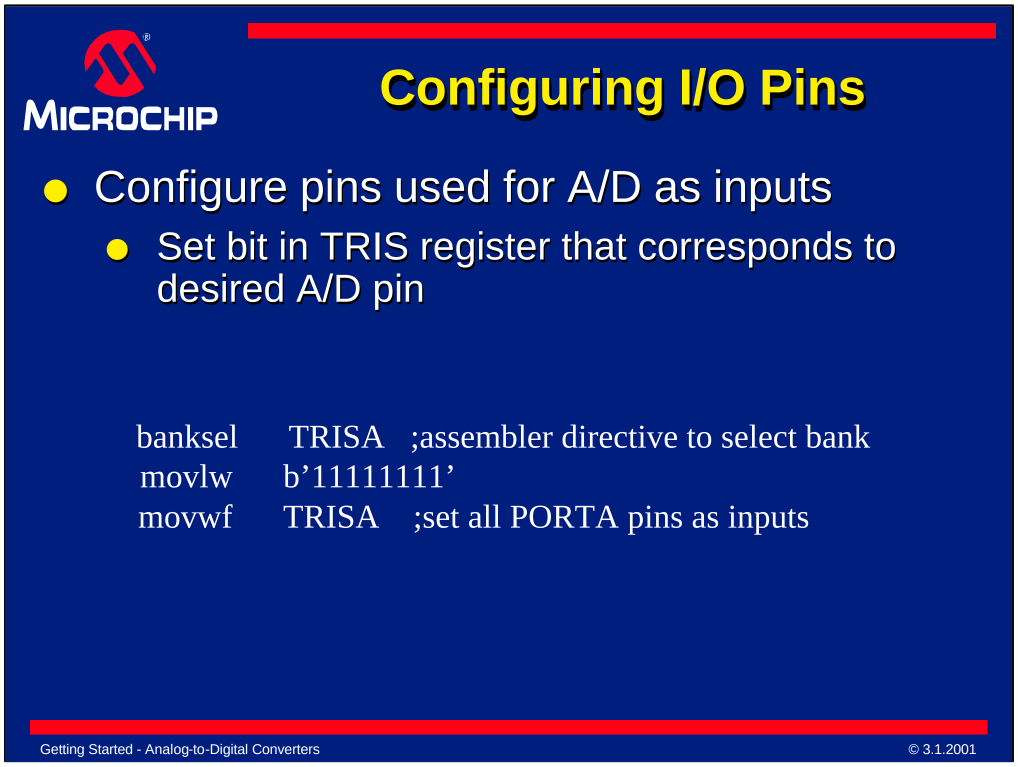
There are multiple sources for the analog-to-digital converter clock. These are the main oscillator frequency divided by 2, 8 or 32, or a dedicated internal RC clock that has a typical period of 4 or 6uS. Since the conversion time is a function of the analog-to-digital converter clock speed, a faster clock will result in a faster conversion time. It must be remembered though, that the typical minimum period for the clock is 1.6 micro seconds. In addition there is a maximum period for the analog-to-digital converter clock. This is typically 10uS. If these specifications are not met, the conversion results will be inaccurate. Consult the analog-to-digital converter section of your device data sheet for the minimum analog-to-digital clock period for your microcontroller.


A/D Conversion Steps

- Configure I/O Pins
- Select the Channel to Convert
- Configure and Enable the A/D
- Wait the Acquisition Time
- Initiate the Conversion
- Wait for the Conversion to Complete
- Read the Result

Getting Started - Analog-to-Digital Converters © 3.1.2001

The following steps are required to set up the analog-to-digital converter and perform a conversion. First, the I/O pins to be used for analog-to-digital conversion are configured as analog inputs. The channel to convert is then selected and the module configured and enabled. A delay for the acquisition time is then provided, after which, the conversion is initiated. Once the conversion has completed, the result can be read. Each step of this process will be explained in greater detail in the following slides.



Configuring I/O Pins

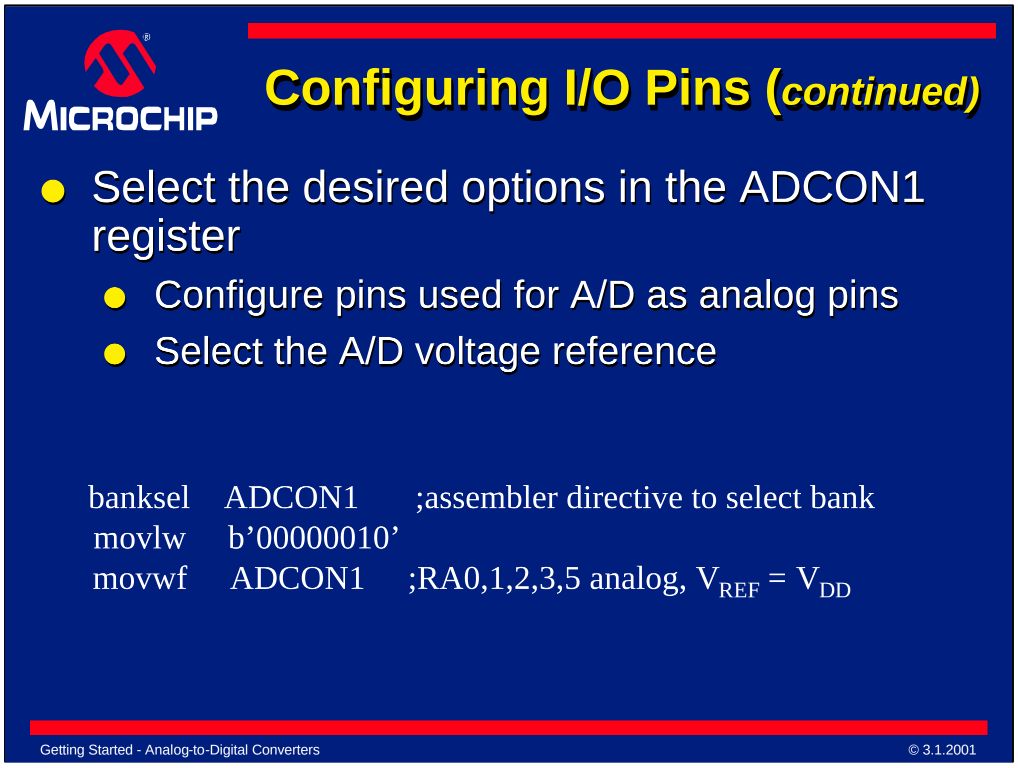
- Configure pins used for A/D as inputs
  - Set bit in TRIS register that corresponds to desired A/D pin

```
banksel     TRISA   ;assembler directive to select bank
movlw       b’11111111’
movwf       TRISA   ;set all PORTA pins as inputs
```

Getting Started - Analog-to-Digital Converters © 3.1.2001

All pins used as analog inputs should be configured by setting the corresponding bit in the TRIS register. The contents of the TRISA register determine whether the pins of PORTA are inputs or outputs.

In this example, all pins of PORTA are to be configured as inputs. The “banksel” is an assembler directive that generates the proper code to select the bank of the specified register, in this case TRISA. A value of FF hexadecimal is then moved into the W register. The contents of the W register are then moved into the TRISA register. This sets all pins of PORTA to inputs.



Configuring I/O Pins (*continued)*

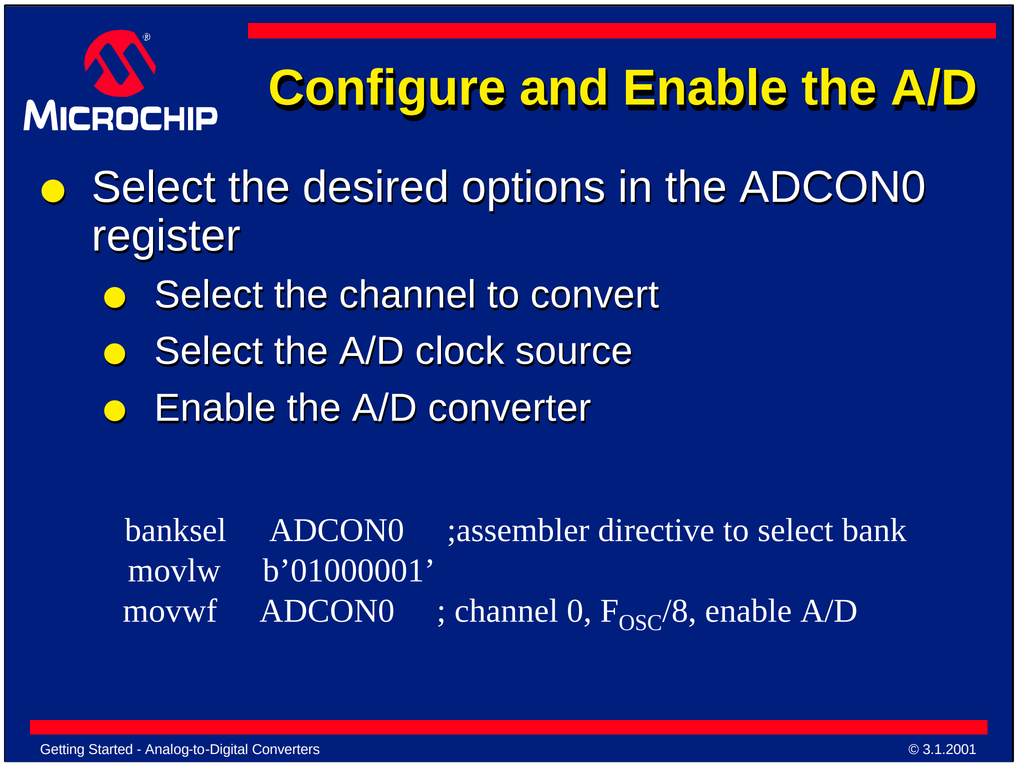
- Select the desired options in the ADCON1 register
  - Configure pins used for A/D as analog pins
  - Select the A/D voltage reference

```
banksel   ADCON1      ;assembler directive to select bank
movlw     b'00000010'
movwf     ADCON1     ;RA0,1,2,3,5 analog, VREF = VDD
```

Getting Started - Analog-to-Digital Converters © 3.1.2001

All pins used for analog-to-digital conversion should also be configured as analog pins. The value in the ADCON1 register determines if pins are configured as analog or digital, and the source of the analog-to-digital converter voltage references. Consult the data sheet of the device for a table that shows the allowable configuration options for ADCON1. Please note that if a pin is configured as digital, and it is used as an analog input, extra current will be consumed anytime it is in a state between the high and low logic levels. On the other hand, if a pin is configured as analog and it is used as a digital pin, it will not read properly because the digital input circuitry for the pin will be disabled.

In this example, pins RA0,1,2,3 and 5 are to be configured as analog pins, and the voltage reference is selected as VDD. The "banksel" is an assembler directive that generates the proper code to select the bank of the specified register, in this case ADCON1. A value of two is then moved into the W register. The value in the W register is then moved into the ADCON1 register, selecting the desired configuration.



Configure and Enable the A/D

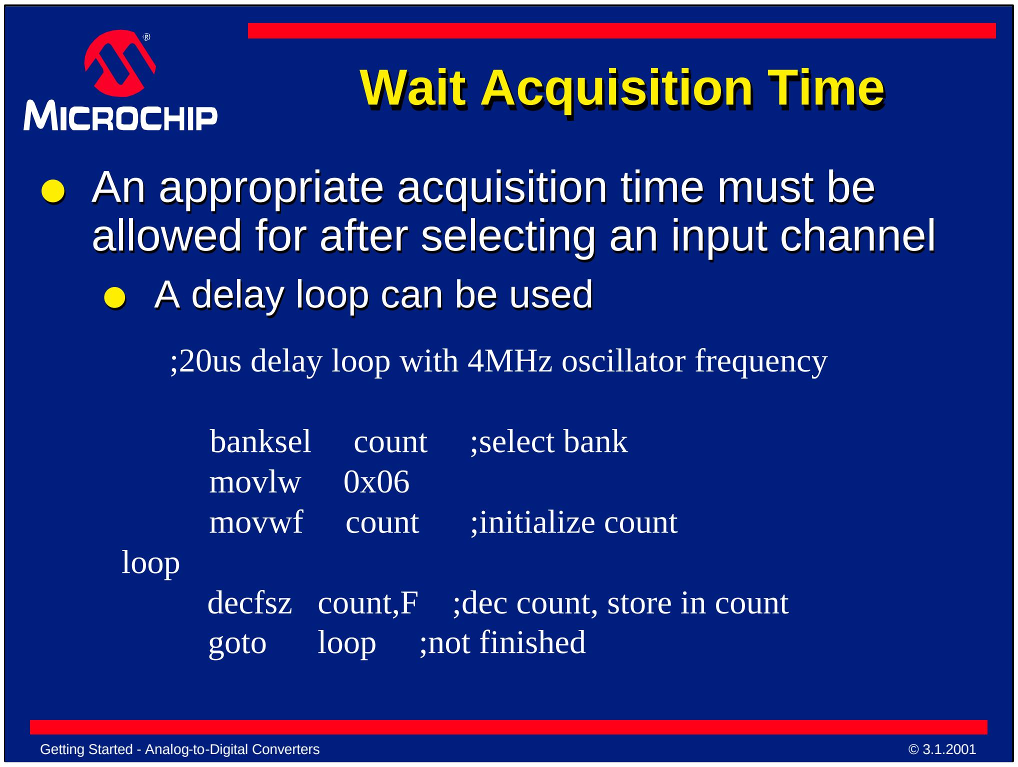
- Select the desired options in the ADCON0 register
  - Select the channel to convert
  - Select the A/D clock source
  - Enable the A/D converter

```
banksel    ADCON0    ;assembler directive to select bank
movlw      b'01000001'
movwf      ADCON0    ; channel 0, FOSC/8, enable A/D
```

Getting Started - Analog-to-Digital Converters © 3.1.2001

The input channel and the analog-to-digital converter clock are selected, and the converter is enabled, by the options selected in the ADCON0 register. Consult the data sheet of your device for a listing of the options in the ADCON0 register.

In this example, the analog-to-digital converter clock is selected as the main oscillator frequency divided by eight, the input channel selected is channel zero, and the converter is being enabled. It is important to note that the converter is being enabled but the conversion is not being started at this time. The conversion should not be initiated in the same instruction that enables the converter, since this will violate the acquisition time requirements. A value of 41 hexadecimal is moved to the W register. The contents of the W register are then moved into the ADCON0 resister, configuring and enabling the analog-to-digital converter.



Wait Acquisition Time

- An appropriate acquisition time must be allowed for after selecting an input channel
  - A delay loop can be used

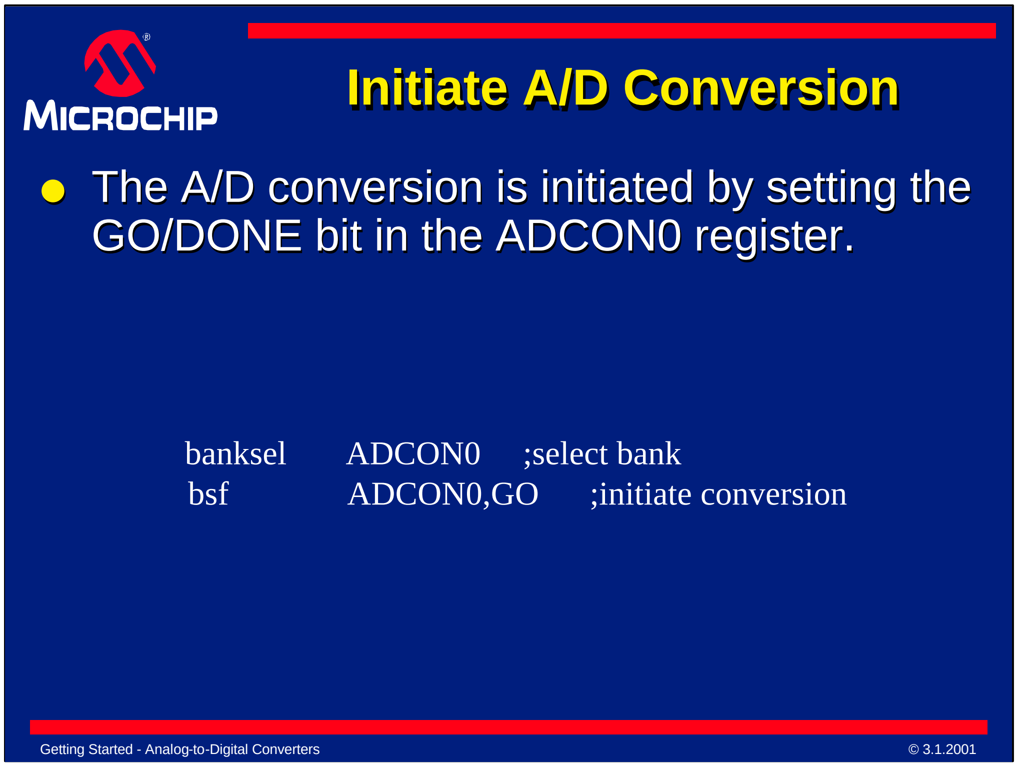
```
;20us delay loop with 4MHz oscillator frequency

        banksel   count    ;select bank
        movlw     0x06
        movwf     count    ;initialize count
loop
        decfsz  count,F   ;dec count, store in count
        goto    loop    ;not finished
```

Getting Started - Analog-to-Digital Converters © 3.1.2001

After a channel is selected in the ADCON0 register, you must allow for sufficient acquisition time before initiating the conversion. An easy way to wait the acquisition time is through the use of a delay loop.

In this example, assume that a minimum acquisition time of 20us is required. Also assume that a register named “count” has been previously declared, and that a 4MHz oscillator is being used. A value of 6 is first moved into the W register. The contents of the W register are then moved to the count register. The actual value loaded into the count register will depend on the required acquisition time and the oscillator frequency. The decfsz instruction decrements the count register and stores the decremented result back in the count register. If the result after the decrement is zero, the next instruction is skipped, otherwise it is executed. The “goto loop” instruction will cause the program to branch back to the” loop” label. In this case, the “goto loop” instruction will be executed the first five times through the loop. The sixth time it will be skipped and the delay loop will have completed. When this delay loop has finished, the required 20us acquisition time will have passed, and the conversion can be initiated.



Initiate A/D Conversion

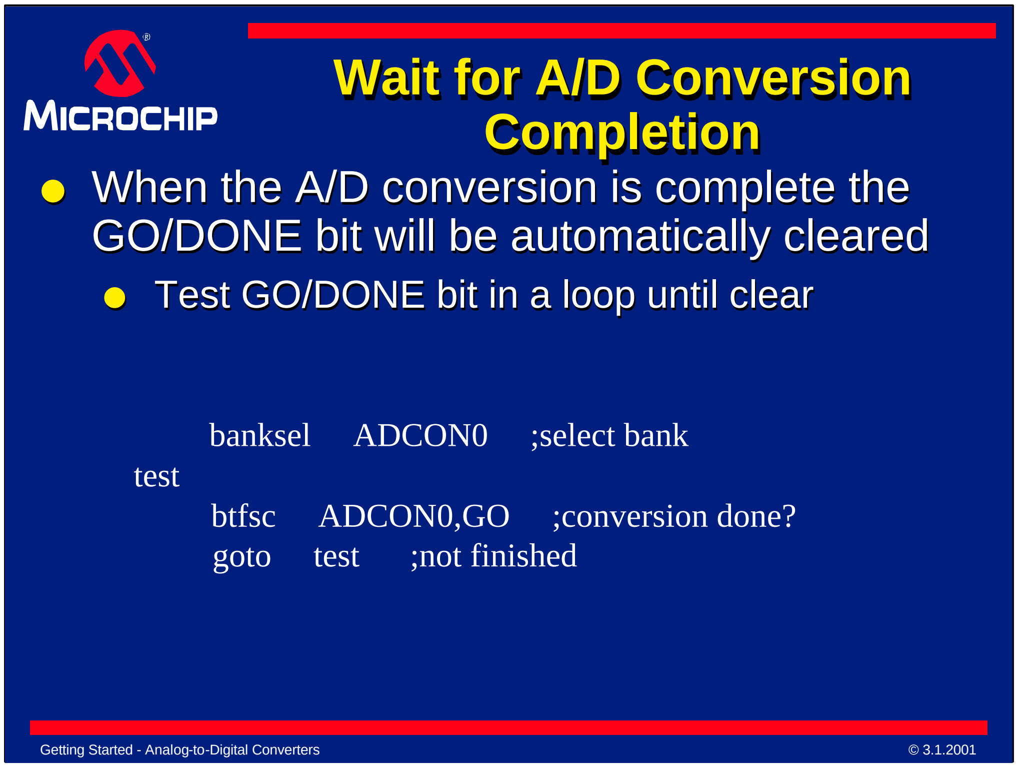
- The A/D conversion is initiated by setting the GO/DONE bit in the ADCON0 register.

```
banksel     ADCON0     ;select bank
bsf         ADCON0,GO     ;initiate conversion
```

Getting Started - Analog-to-Digital Converters © 3.1.2001

After the acquisition time has passed, the conversion can be started. This is accomplished by setting the GO bit in the ADCON0 register.

In this example, the “bsf” instruction is used to set the GO bit in the ADCON0 register. This starts the analog-to-digital conversion process.



Wait for A/D Conversion Completion

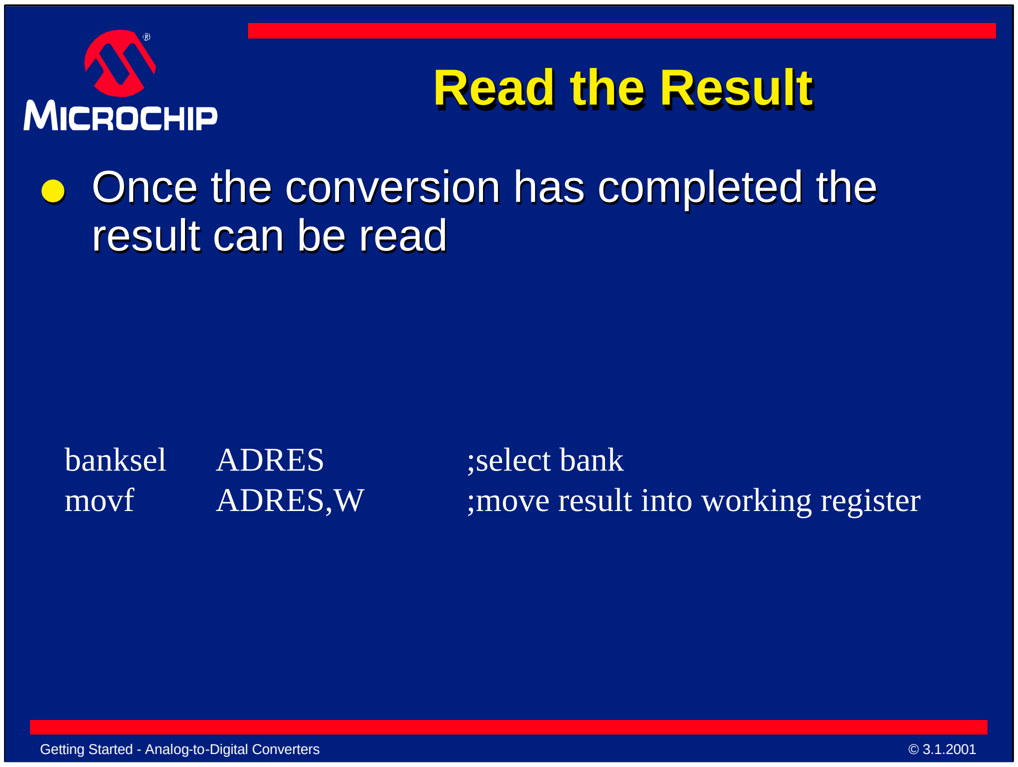
- When the A/D conversion is complete the GO/DONE bit will be automatically cleared
  - Test GO/DONE bit in a loop until clear

```
        banksel    ADCON0     ;select bank
test
        btfsc    ADCON0,GO     ;conversion done?
        goto     test      ;not finished
```

Getting Started - Analog-to-Digital Converters © 3.1.2001

Once the conversion has been initiated, the conversion time must be allowed for before reading the result. The status of the analog-to-digital conversion is indicated by the GO/DONE bit in the ADCON0 register, which is automatically cleared when the conversion is complete. A polling loop can be used to wait for the GO/DONE bit to be cleared, thus indicating that the conversion is complete.

In this example, a "btfsc" instruction tests the GO/DONE bit of the ADCON0 register. If this bit is set, the next instruction which is the "goto test", is executed and the program loops back to the bit test instruction. Once the GO/DONE bit is clear, the "goto test" instruction will be skipped and program execution can continue.



Read the Result

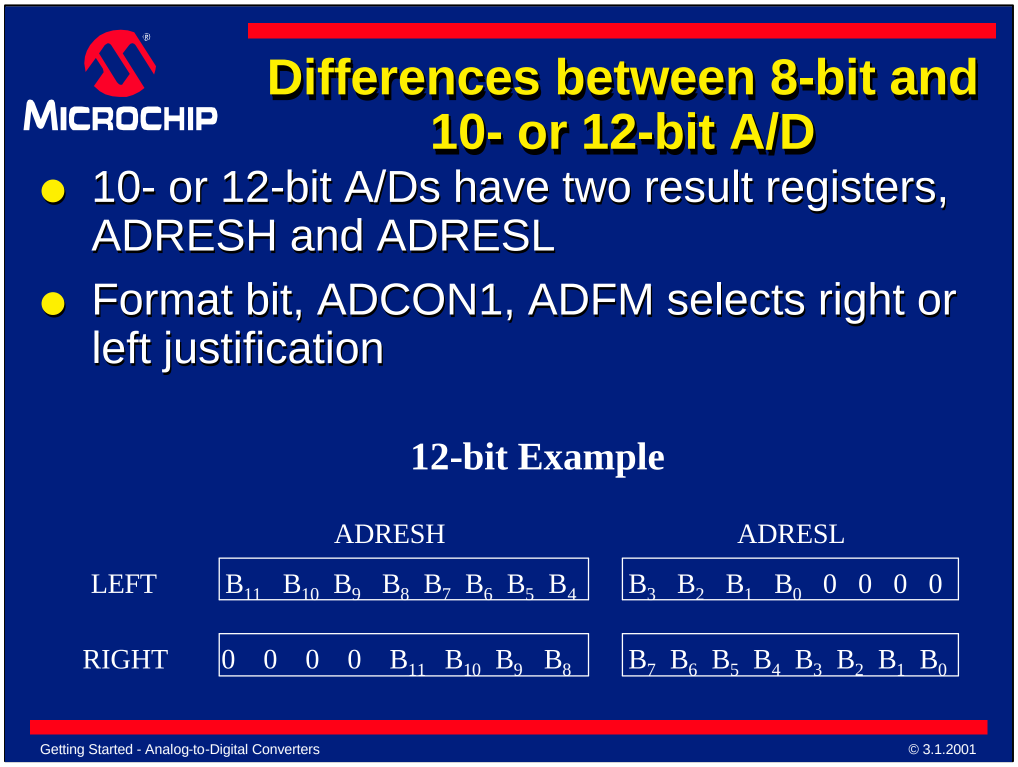
- Once the conversion has completed the result can be read

```
banksel     ADRES           ;select bank
movf        ADRES,W         ;move result into working register
```

Getting Started - Analog-to-Digital Converters © 3.1.2001

Once the analog-to-digital conversion has completed, the result can be read. The result of the conversion is automatically placed in the ADRES register upon completion.

In this example, the contents of the ADRES register are moved into the W register for further processing by the user’s application.



Differences between 8-bit and 10- or 12-bit A/D

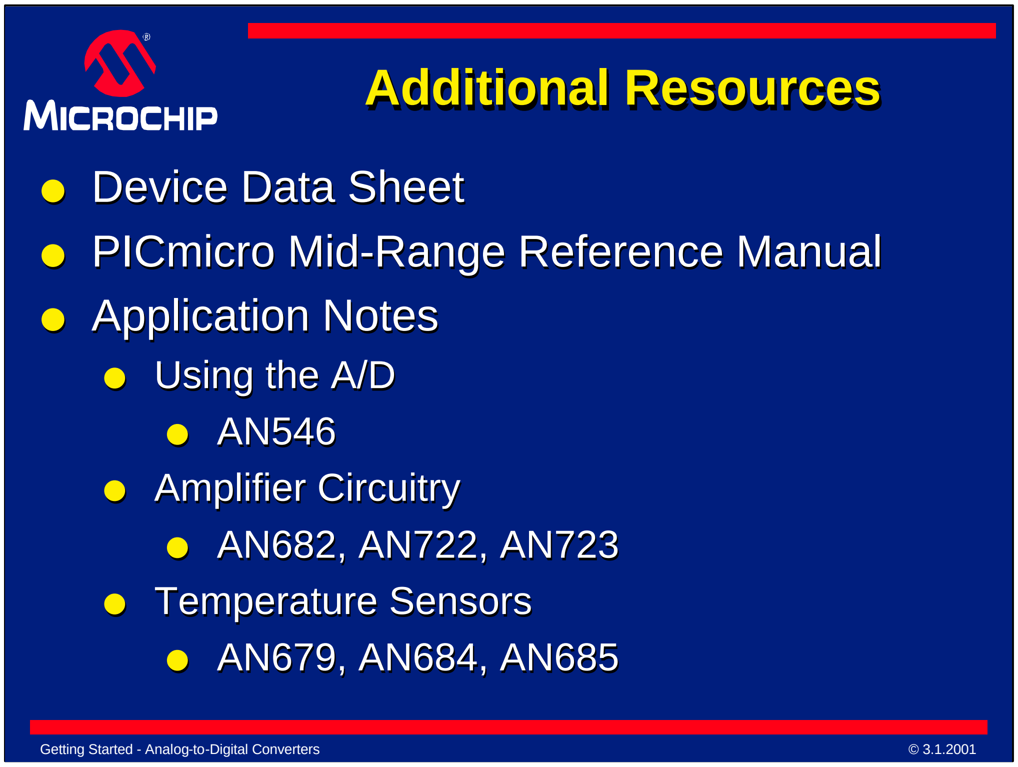
- 10- or 12-bit A/Ds have two result registers, ADRESH and ADRESL
- Format bit, ADCON1, ADFM selects right or left justification

12-bit Example

	ADRESH	ADRESL
LEFT	B_{11} B_{10} B_9 B_8 B_7 B_6 B_5 B_4	B_3 B_2 B_1 B_0 0 0 0 0
RIGHT	0 0 0 0 B_{11} B_{10} B_9 B_8	B_7 B_6 B_5 B_4 B_3 B_2 B_1 B_0

Getting Started - Analog-to-Digital Converters © 3.1.2001

Since our midrange microcontrollers have an 8-bit data width, the result of a 10- or 12-bit analog-to-digital conversion must be represented in two registers. The high and low result registers are ADRESH and ADRESL. The result format bit is ADFM of the ADCON1 register. The result can be either left or right justified. When the result is left justified, the eight most significant bits will be placed in the ADRESH register with the least significant bits placed in the ADRESL register. When right justified, the eight least significant bits will be placed in the ADRESL register with the most significant bits placed in the ADRESH register. The selection of right or left justification will depend on the requirements of the user's software.

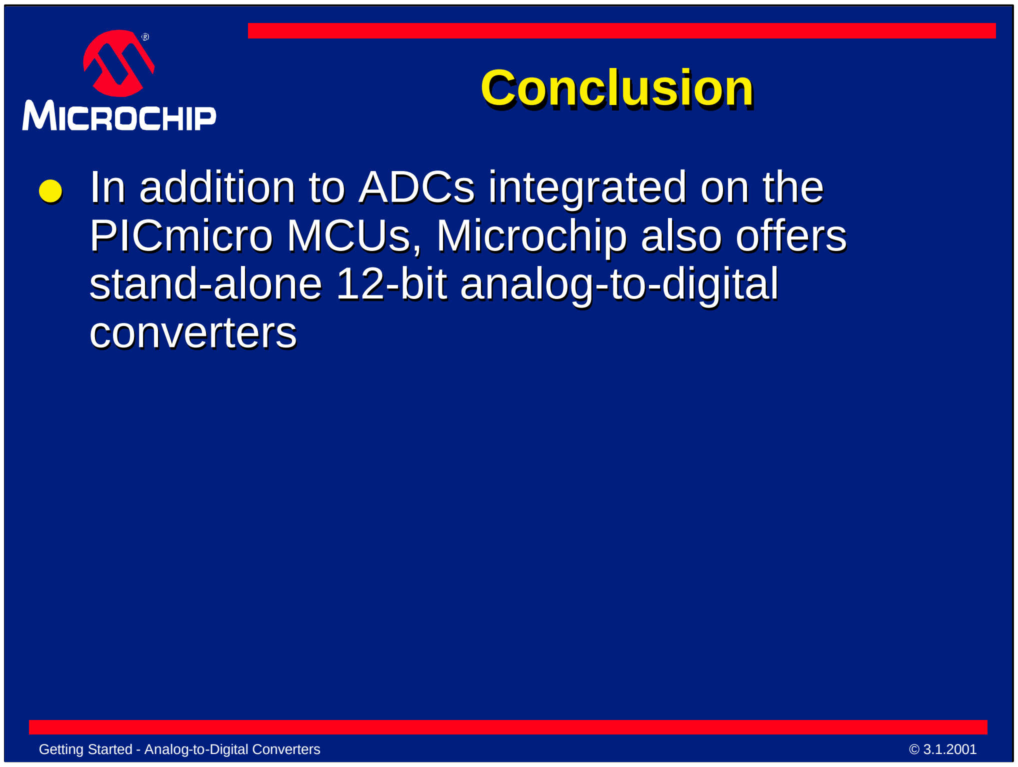


Additional Resources

- Device Data Sheet
- PICmicro Mid-Range Reference Manual
- Application Notes
  - Using the A/D
    - AN546
  - Amplifier Circuitry
    - AN682, AN722, AN723
  - Temperature Sensors
    - AN679, AN684, AN685

Getting Started - Analog-to-Digital Converters © 3.1.2001

We have a number of additional resources with useful information on the analog-to-digital converter peripheral and topics related to its use. First, obtain the data sheet for the device you have selected before beginning any PICmicro MCU design. Data sheets for our microcontrollers can be obtained under the “PICmicro MCU” section of our website and data sheet provide specifications and information specific to each particular device. Also available in this section of the website is the *“PICmicro Mid-Range Reference Manual”* which provides additional general information on the use of our mid-range devices. The “Application Notes” section of our web site contains a number of documents that pertain to the analog-to-digital converter as well as other related topics. AN546, *Using the Analog to Digital Converter*, is a general application note on the use of the analog-to-digital converter peripheral. AN682, *Using Single Supply Operational Amplifiers in Embedded Systems*; AN722, *Operational Amplifier Topologies and DC Specifications*, and AN723, *Operational Amplifier AC Specifications and Applications*, discuss amplifier circuits and considerations. Finally, AN679, *Temperature Sensing Technologies*, AN684, *Single Supply Temperature Sensing with Thermocouples*, and AN685, *Thermistors in Single Supply Temperature Sensing Circuits*, discuss some temperature sensor applications.



Conclusion

- In addition to ADCs integrated on the PICmicro MCUs, Microchip also offers stand-alone 12-bit analog-to-digital converters

Getting Started - Analog-to-Digital Converters © 3.1.2001

This concludes a brief examination of the analog-to-digital converter peripheral found on Microchip’s PICmicro devices.

In addition to the on-chip ADCs, Microchip also manufactures the stand-alone MCP320X 12-bit analog-to-digital converter family. These stand-alone peripherals feature a successive approximation register (SAR) architecture, and an industry-standard SPI™ bus interface. Devices are available with 1, 2, 4, or 8 input channels and are offered in PDIP, SOIC and TSSOP packages. Applications for the MCP320X family include data acquisition, instrumentation and measurement, multi-channel data loggers, industrial PCs, motor control, robotics, industrial automation, smart sensors, portable instrumentation, and home medical appliances Watch for future tutorials on these products or visit the Analog/Interface area of the Microchip website for Datasheets, Application Notes, Seminar and Workshop schedules, and other helpful information.

MICROCHIP®