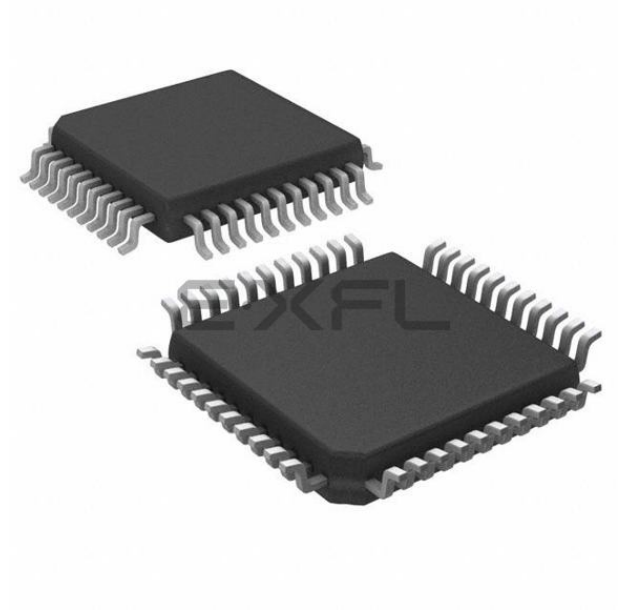


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

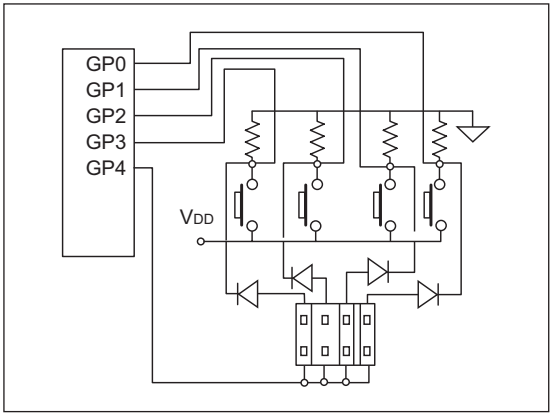
Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	4MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	33
Program Memory Size	7KB (4K x 14)
Program Memory Type	OTP
EEPROM Size	-
RAM Size	192 x 8
Voltage - Supply (Vcc/Vdd)	4V ~ 5.5V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	44-QFP
Supplier Device Package	44-MQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic16c65b-04i-pq

TIP #9 Decode Keys and ID Settings

Buttons and jumpers can share I/O’s by using another I/O to select which one is read. Both buttons and jumpers are tied to a shared pull-down resistor. Therefore, they will read as ‘0’ unless a button is pressed or a jumper is connected. Each input (GP3/2/1/0) shares a jumper and a button. To read the jumper settings, set GP4 to output high and each connected jumper will read as ‘1’ on its assigned I/O or ‘0’ if it’s not connected. With GP4 output low, a pressed button will be read as ‘1’ on its assigned I/O and ‘0’ otherwise.

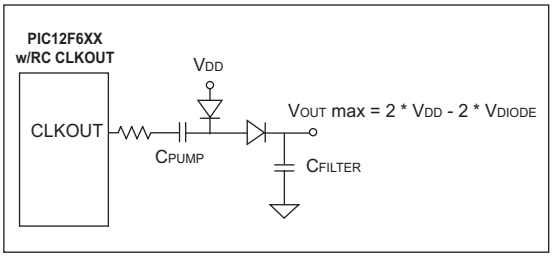
Figure 9-1



- When GP4 = 1 and no keys are pressed, read ID setting
- When GP4 = 0, read the switch buttons

TIP #10 Generating High Voltages

Figure 10-1



Voltages greater than V_{DD} can be generated using a toggling I/O. PIC MCUs CLKOUT/OSC2 pin toggles at one quarter the frequency of OSC1 when in external RC oscillator mode. When OSC2 is low, the V_{DD} diode is forward biased and conducts current, thereby charging C_{PUMP}. After OSC2 is high, the other diode is forward biased, moving the charge to C_{FILTER}. The result is a charge equal to twice the V_{DD} minus two diode drops. This can be used with a PWM, a toggling I/O or other toggling pin.

TIP #8 Modulation Formats

The CCP module, configured in Compare mode, can be used to generate a variety of modulation formats. The following figures show four commonly used modulation formats:

Figure 8-1: Pulse-width Modulation

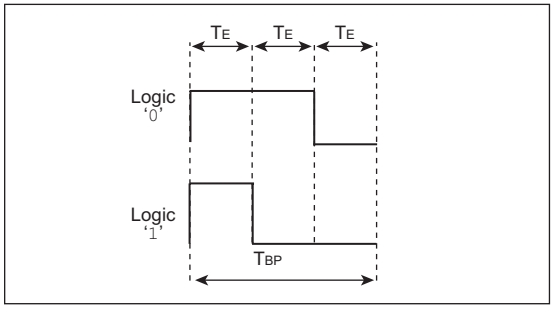


Figure 8-2: Manchester

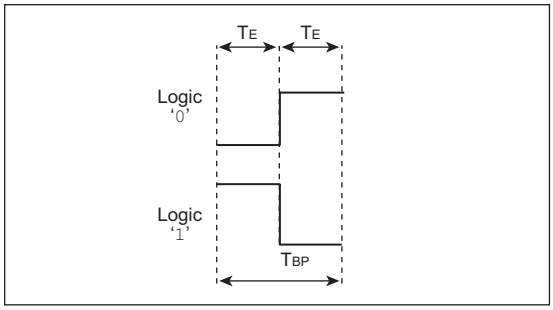


Figure 8-3: Pulse Position Modulation

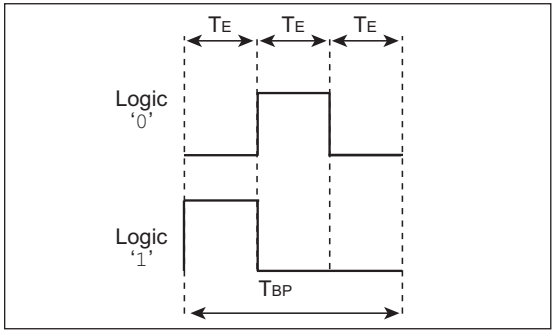
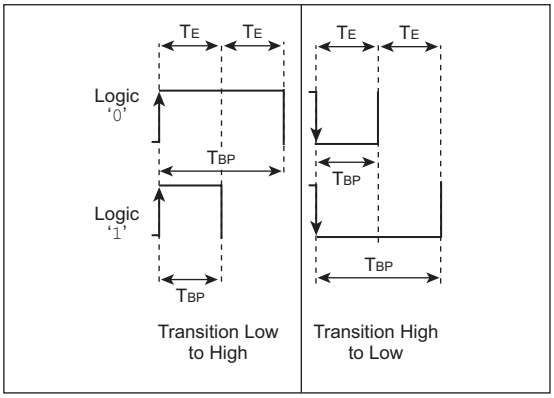


Figure 8-4: Variable Pulse-width Modulation



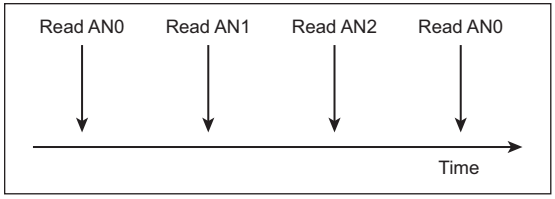
The figures show what a logic '0' or a logic '1' looks like for each modulation format. A transmission typically resembles an asynchronous serial transmission consisting of a Start bit, followed by 8 data bits, and a Stop bit.

T_E is the basic timing element in each modulation format and will vary based on the desired baud rate.

Trigger Special Event mode can be used to generate T_E, (the basic timing element). When the CCPx interrupt is generated, code in the ISR routine would implement the desired modulation format (additional modulation formats are also possible).

TIP #11 Sequential ADC Reader

Figure 11-1: Timeline



Trigger Special Event mode (a sub-mode in Compare mode) generates a periodic interrupt in addition to automatically starting an A/D conversion when Timer1 matches CCPRxL and CCPRxH. The following example problem demonstrates how to sequentially read the A/D channels at a periodic interval.

Example

Given the PIC16F684 running on its 8 MHz internal oscillator, configure the microcontroller to sequentially read analog pins AN0, AN1 and AN2 at 30 ms intervals.

Step #1: Determine Timer1 Prescaler

a) Timer1 overflows at: Tosc*4*65536* prescaler.

b) For a prescaler of 1:1, the Timer1 overflow occurs in 32.8 ms.

c) This is greater than 30 ms, so a prescaler of 1 is adequate.

Step #2: Calculate CCPR1 (CCPR1L and CCPR1H)

a) CCPR1 = Interval Time/(Tosc*4*prescaler) = 0.030/(125 ns*4*1) = 6000 = 0xEA60

b) Therefore, CCPR1L = 0x60, and CCPR1H = 0xEA

Step #3: Configuring CCP1CON

The ECCP module should be configured in Trigger Special Event mode. This mode generates an interrupt when Timer1 equals the value specified in CCPR1. Timer1 is automatically cleared and the GO bit in ADCON0 is automatically set. For this mode, CCP1CON = '`b00001011`'.

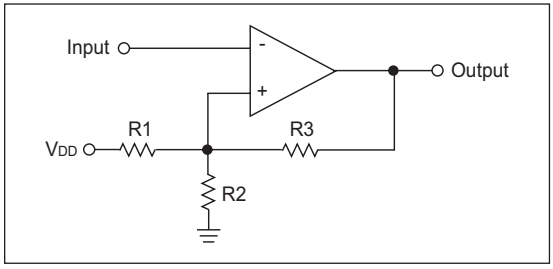
Step #4: Add Interrupt Service Routine Logic

When the ECCP interrupt is generated, select the next A/D pin for reading by altering the ADCON0 register.

TIP #3 Hysteresis

When the voltages on a comparator's input are nearly equal, external noise and switching noise from inside the microcontroller can cause the comparator output to oscillate or "chatter." To prevent chatter, some of the comparator output voltage is fed back to the non-inverting input of the comparator to form hysteresis (see Figure 3-1). Hysteresis moves the comparator threshold up when the input is below the threshold, and down when the input is above the threshold. The result is that the input must overshoot the threshold to cause a change in the comparator output. If the overshoot is greater than the noise present on the input, the comparator output will not chatter.

Figure 3-1: Comparator with Hysteresis



To calculate the resistor values required, first determine the high and low threshold values which will prevent chatter (V_{TH} and V_{TL}). Using V_{TH} and V_{TL}, the average threshold voltage can be calculated using the equation.

Equation 3-1

$$V_{AVG} = \frac{V_{DD} * V_{TL}}{V_{DD} - V_{TH} + V_{TL}}$$

Next, choose resistor values that satisfy Equation 3-2 and calculate the equivalent resistance using Equation 3-3.

Note: A continuous current will flow through R1 and R2. To limit the power dissipation in R1 and R2 the total resistance of R1 and R2 should be at least 1k. The total resistance of R1 and R2 should also be kept below 10K to keep the size of R3 small. Large values for R3, 100k-10 M , can produce voltage offsets at the non-inverting input due to the comparator's input bias current.

Equation 3-2

$$V_{AVG} = \frac{V_{DD} * R2}{R1 + R2}$$

Equation 3-3

$$R_{EQ} = \frac{R1 * R2}{R1 + R2}$$

Then, determine the feedback divider ratio D_R, using Equation 3-4.

Equation 3-4

$$D_R = \frac{(V_{TH} - V_{TL})}{V_{DD}}$$

Finally, calculate the feedback resistor R3 using Equation 3-5.

Equation 3-5

$$R3 = R_{EQ} \left[\left(\frac{1}{D_R} \right) - 1 \right]$$

Example:

- A V_{DD} = 5.0V, V_H = 3.0V and V_L = 2.5V
- V_{AVG} = 2.77V
- R = 8.2k and R2 = 10k, gives a V_{AVG} = 2.75V
- R_{EQ} = 4.5k
- D_R = .1
- R3 = 39k (40.5 calculated)
- V_{HACT} = 2.98V
- V_{LACT} = 2.46V

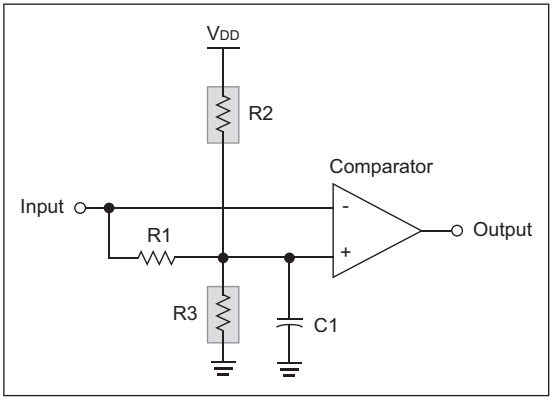
TIP #6 Data Slicer

In both wired and wireless data transmission, the data signal may be subject to DC offset shifts due to temperature shifts, ground currents or other factors in the system. When this happens, using a simple level comparison to recover the data is not possible because the DC offset may exceed the peak-to-peak amplitude of the signal. The circuit typically used to recover the signal in this situation is a data slicer.

The data slicer shown in Figure 6-1 operates by comparing the incoming signal with a sliding reference derived from the average DC value of the incoming signal. The DC average value is found using a simple RC low-pass filter (R1 and C1). The corner frequency of the RC filter should be high enough to ignore the shifts in the DC level while low enough to pass the data being transferred.

Resistors R2 and R3 are optional. They provide a slight bias to the reference, either high or low, to give a preference to the state of the output when no data is being received. R2 will bias the output low and R3 will bias the output high. Only one resistor should be used at a time, and its value should be at least 50 to 100 times larger than R1.

Figure 6-1: Data Slicer



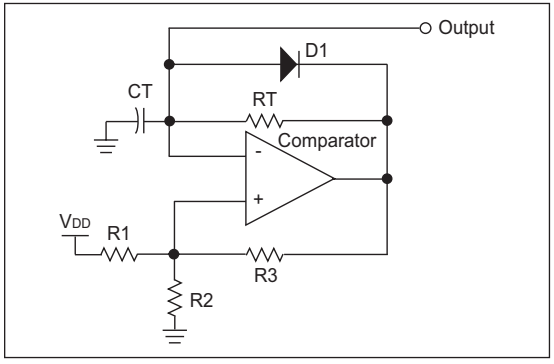
Example:

Data rate of 10 kbits/second. A low pass filter frequency of 500 Hz: R1 = 10k, C1 = 33 μF. R2 or R3 should be 500k to 1 MB.

TIP #9 Multi-Vibrator (Ramp Wave Output)

A multi-vibrator (ramp wave output) is an oscillator designed around a voltage comparator or operational amplifier that produces an asymmetrical output waveform (see Figure 9-1). Resistors R1 through R3 form a hysteresis feedback path from the output to the non-inverting input. Resistor RT, diode D1 and capacitor CT form a time delay network between the output and the inverting input. At the start of the cycle, CT is discharged holding the non-inverting input at ground, forcing the output high. A high output forces the non-inverting input to the high threshold voltage (see Tip #3) and charges CT through RT. When the voltage across CT reaches the high threshold voltage, the output is forced low. A low output drops the non-inverting input to the low threshold voltage and discharges CT through D1. Because the dynamic on resistance of the diode is significantly lower than RT, the discharge of CT is small when compared to the charge time, and the resulting waveform across CT is a pseudo ramp function with a ramping charge phase and a short-sharp discharge phase.

Figure 9-1: Ramp Waveform Multi-Vibrator



To design this multi-vibrator, first design the hysteresis feedback path using the procedure in Tip #3. Remember that the peak-to-peak amplitude of the ramp wave will be determined by the hysteresis limits. Also, be careful to choose threshold voltages (V_{TH} and V_{TL}) that are evenly spaced within the common mode range of the comparator.

Then use V_{TH} and V_{TL} to calculate values for RT and CT that will result in the desired oscillation frequency F_{OSC}. Equation 9-1 defines the relationship between RT, CT, V_{TH}, V_{TL} and F_{OSC}.

Equation 9-1

$$F_{OSC} = \frac{1}{RT * CT * \ln(V_{TH}/V_{TL})}$$

This assumes that the dynamic on resistance of D1 is much less than RT.

Example:

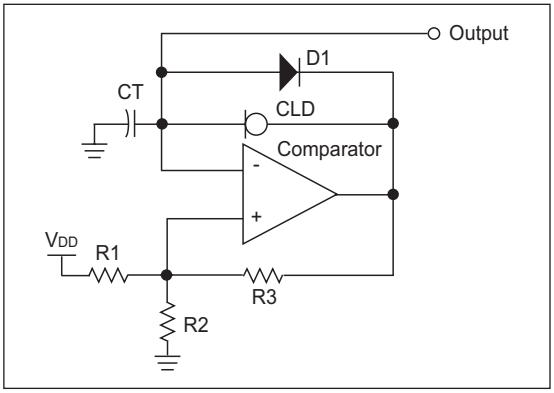
- V_{DD} = 5V, V_{TH} = 1.666V and V_{TH} = 3.333V
- R1, R2 and R3 = 10k
- R_T = 15k, C_T = .1 µF for a F_{OSC} = 906 Hz

Note: Replacing R_T with a current limiting diode will significantly improve the linearity of the ramp wave form. Using the example shown above, a CCL1000 (1 mA Central Semiconductor CLD), will produce a very linear 6 kHz output (see Equation 9-2).

Equation 9-2

$$F_{OSC} = \frac{I_{CLD}}{C\,(V_{TH} - V_{TL})}$$

Figure 9-2: Alternate Ramp Waveform Multi-Vibrator Using a CLD



TIP #12 Making an Op Amp Out of a Comparator

When interfacing to a sensor, some gain is typically required to match the full range of the sensor to the full range of an ADC. Usually this is done with an operational amplifier, however, in cost sensitive applications, an additional active component may exceed the budget. This tip shows how an on-chip comparator can be used as an op amp like gain stage for slow sensor signals. Both an inverting and non-inverting topology are shown (see Figure 12-1 and Figure 12-2).

Figure 12-1: Non-Inverting Amplifier

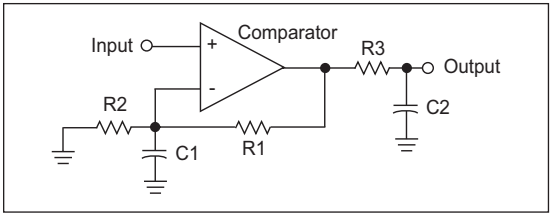
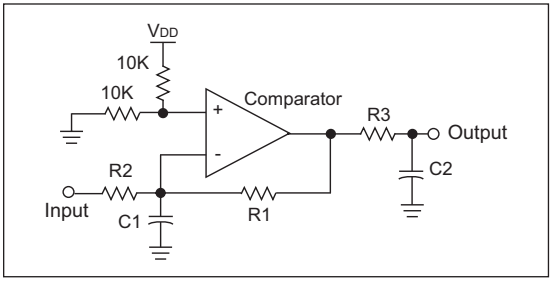


Figure 12-2: Inverting Amplifier



To design a non-inverting amplifier, choose resistors R1 and R2 using the Gain formula for an op amp non-inverting amplifier (see Equation 12-1).

Equation 12-1

$$\text{Gain} = \frac{R1 + R2}{R2}$$

Once the gain has been determined, values for R3 and C2 can be determined. R3 and C2 form a low-pass filter on the output of the amplifier. The corner frequency of the low pass should be 2 to 3 times the maximum frequency of the signal being amplified to prevent attenuation of the signal, and R3 should be kept small to minimize the output impedance of the amplifier. Equation 12-2 shows the relationship between R3, C2 and the corner frequency of the low pass filter.

Equation 12-2

$$F_{CORNER} = \frac{1}{2 * \ * R3 * C2}$$

A value for C1 can then be determined using Equation 12-3. The corner frequency should be the same as Equation 12-3.

Equation 12-3

$$F_{CORNER} = \frac{1}{2 * \ * (R1 \text{ II } R2) * C2}$$

To design an inverting amp, choose resistors R1 and R2 using the Gain formula for an op amp inverting amplifier (see Equation 12-4).

Equation 12-4

$$\text{Gain} = \frac{R1}{R2}$$

Then choose values for the resistor divider formed by R4 and R5. Finally choose C1 and C2 as shown in the non-inverting amplifier design.

Example:

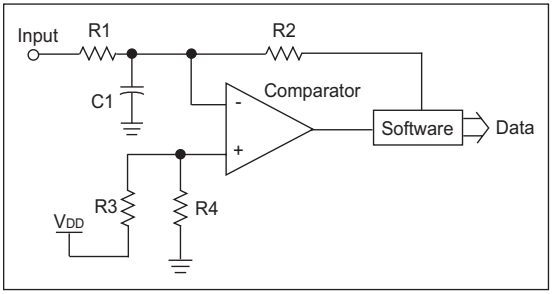
- For C2 will set the corner F
- Gain = 6.156, R1 = R3 = 19.8k
- R2 = 3.84k, C1 = .047 μF, F_{CORNER} = 171 Hz
- C2 = .22 μF

TIP #14 Delta-Sigma ADC

This tip describes the creation of a hardware/software-based Delta-Sigma ADC. A Delta-Sigma ADC is based on a Delta-Sigma modulator composed of an integrator, a comparator, a clock sampler and a 1-bit DAC output. In this example, the integrator is formed by R1 and C1. The comparator is an on-chip voltage comparator. The clock sampler is implemented in software and the 1-bit DAC output is a single I/O pin. The DAC output feeds back into the integrator through R2.

Resistors R3 and R4 form a $V_{DD}/2$ reference for the circuit (see Figure 14-1).

Figure 14-1: Delta-Sigma Modulator



In operation, the feedback output from the software is a time sampled copy of the comparator output. In normal operation, the modulator output generates a PWM signal which is inversely proportional to the input voltage. As the input voltage increases, the PWM signal will drop in duty cycle to compensate. As the input decreases, the duty cycle rises.

To perform an A-to-D conversion, the duty cycle must be integrated over time, digitally, to integrate the duty cycle to a binary value. The software starts two counters. The first counts the total number of samples in the conversion and the second counts the number of samples that were low. The ratio of the two counts is equal to the ratio of the input voltage over V_{DD}.

> **Note:** This assumes that R1 and R2 are equal and R3 is equal to R4. If R1 and R2 are not equal, then the input voltage is also scaled by the ratio of R2 over R1, and R3 must still be equal to R4.

For a more complete description of the operation of a Delta-Sigma ADC and example firmware, see Application Note AN700 “*Make A Delta-Sigma Converter Using a Microcontroller’s Analog Comparator Module*.”

Example:

- R3 = R4 = 10 kHz
- R1 = R2 = 5.1k
- C1 = 1000 pF

TIP #19 Logic: XOR/XNOR Gate

This tip shows the use of the comparator to implement an XOR gate and its complement the XNOR gate.

The operation is best described in three sections:

1. Both A and B inputs are low
   With both inputs low, the inverting input is held at .7V and the non-inverting is held at ground. This combination results in a low output.
2. Both A and B inputs are high
   With both inputs high, the inverting input is pulled up to V_{DD} and the non-inverting input is equal to 2/3 V_{DD} (the average of V_{DD} inputs and G_{ND}). This combination also results in a low output.
3. Input A or B is high
   With one input high and one low, The inverting input is held at .7V and the non-inverting input is equal to 1/3 V_{DD} (the average of a V_{DD} input and G_{ND}). This combination results in a high output.

Note: Typical propagation delay for the circuit is 250-350 ns using the typical on-chip comparator peripheral of a microcontroller. Delay measurements were made with 10k resistance values.

While the circuit is fairly simple, there are a few requirements for correct operation:

1. The inputs A and B must drive from ground to V_{DD} for the circuit to operate properly.
2. All resistances on the both inputs react with the input capacitance of the comparator, so the speed of the gate will be affected by the source resistance of A and B, as well as, the size of resistors R1, R2, R3 and R4.
3. Resistor R1, R2 and R3 must be equal.
4. Resistor R4 must be small enough to produce a 1.0V, or lower, voltage drop across D1 and D2.

Figure 19-1: XOR Gate

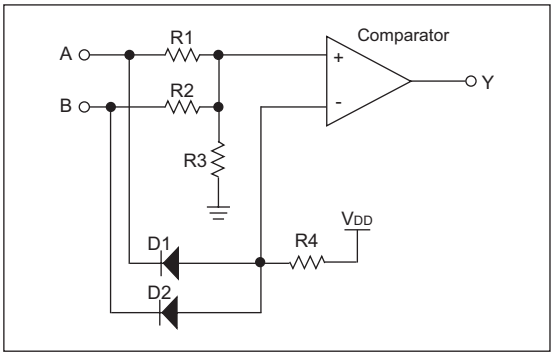
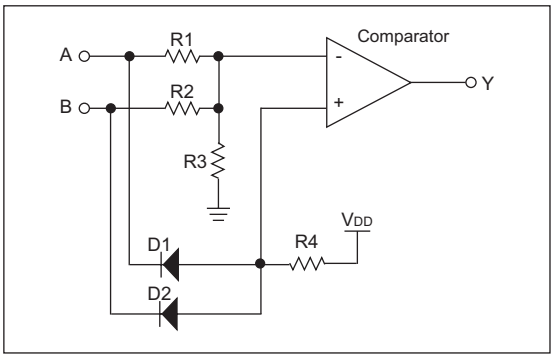


Figure 19-2: XNOR Gate



Example:

- D1 = D2, = 1N4148
- R4 = 10k, R1 = R2 = R3 = 5.1k

TIP #5 Writing a PWM Value to the the CCP Registers With a Mid-Range PIC® Microcontroller

The two PWM LSb’s are located in the CCPCON register of the CCP. This can make changing the PWM period frustrating for a developer. Example 5-1 through Example 5-3 show three macros written for the mid-range product family that can be used to set the PWM period. The first macro takes a 16-bit value and uses the 10 MSb’s to set the PWM period. The second macro takes a 16-bit value and uses the 10 LSb’s to set the PWM period. The last macro takes 8 bits and sets the PWM period. This assumes that the CCP is configured for no more than 8 bits.

Example 5-1: Left Justified 16-Bit Macro

```
pwm_tmp     equ xxx   ;this variable must be
                      ;allocated someplace
setPeriod   macro a   ;a is 2 SFR’s in
                      ;Low:High arrangement
                      ;the 10 MSb’s are the
                      ;desired PWM value
 RRF        a,w       ;This macro will
                      ;change w
 MOVWF      pwm_tmp
 RRF        pwm_tmp,w
 ANDLW      0x30
 IORLW      0x0F
 MOVWF      CCP1CON
 MOVF       a+1,w
 MOVWF      CCPR1L
```

Example 5-2: Right Justified 16-Bit Macro

```
pwm_tmp     equ xxx   ;this variable must be
                      ;allocated someplace
setPeriod   macro a   ;a is 2 bytes in
                      ;Low:High arrangement
                      ;the 10 LSb’s are the
                      ;desired PWM value
 SWAPF      a,w       ;This macro will
                      ;change w
 ANDLW      0x30
 IORLW      0x0F
 MOVWF      CCP1CON
 RLF        a,w
 IORLW      0x0F
 MOVWF      pwm_tmp
 RRF        pwm_tmp,f
 RRF        pwm_tmp,w
 MOVWF      CCPR1L
```

Example 5-3: 8-Bit Macro

```
pwm_tmp     equ xxx   ;this variable must be
                      ;allocated someplace
setPeriod   macro a   ;a is 1 SFR
 SWAPF      a,w       ;This macro will
                      ;change w
 ANDLW      0x30
 IORLW      0x0F
 MOVWF      CCP1CON
 RRF        a,w
 MOVWF      pwm_tmp
 RRF        pwm_tmp,w
 MOVWF      CCPR1L
```

TIP #6 Current Sensing

The torque of an electric motor can be monitored and controlled by keeping track of the current flowing through the motor. Torque is directly proportional to the current. Current can be sensed by measuring the voltage drop through a known value resistor or by measuring the magnetic field strength of a known value inductor. Current is generally sensed at one of two places, the supply side of the drive circuit (high side current sense) or the sink side of the drive circuit (low side current sense). Low side sensing is much simpler but the motor will no longer be grounded, causing a safety issue in some applications. High side current sensing generally requires a differential amplifier with a common mode voltage range within the voltage of the supply.

Figure 6-1: Resistive High Side Current Sensing

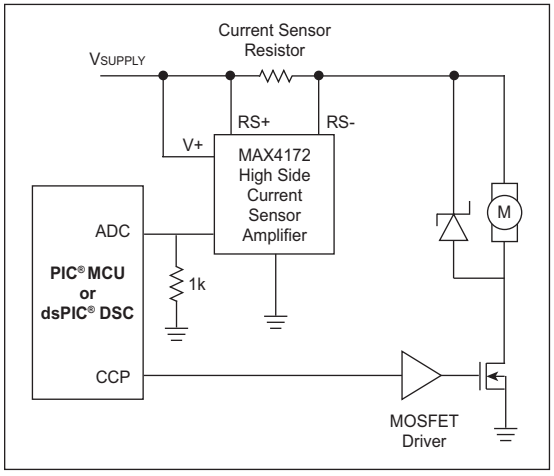
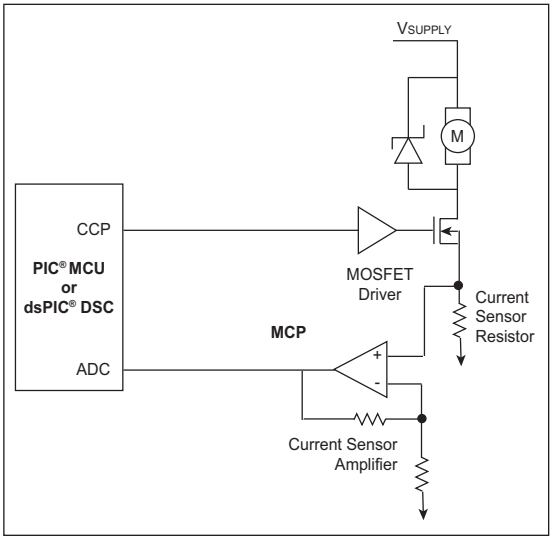
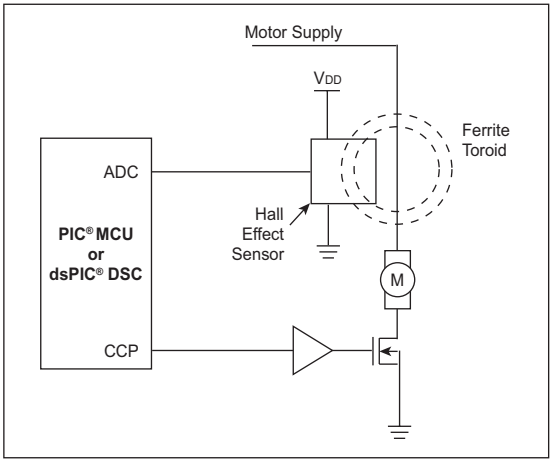


Figure 6-2: Resistive Low Side Current Sensing



Current measurement can also be accomplished using a Hall effect sensor to measure the magnetic field surrounding a current carrying wire. Naturally, this Hall effect sensor can be located on the high side or the low side of the load. The actual location of the sensor does not matter because the sensor does not rely upon the voltage on the wire. This is a non-intrusive method that can be used to measure motor current.

Figure 6-3: Magnetic Current Sensing



Application Note References

- AN220, “*Watt-Hour Meter Using PIC16C923 and CS5460*” (DS00220)
- AN582, “*Low-Power Real-Time Clock*” (DS00582)
- AN587, “*Interfacing PIC® MCUs to an LCD Module*” (DS00587)
- AN649, “*Yet Another Clock Featuring the PIC16C924*” (DS00649)
- AN658, “*LCD Fundamentals Using PIC16C92X Microcontrollers*” (DS00658)
- TB084, “*Contrast Control Circuits for the PIC16F91X*” (DS91084)

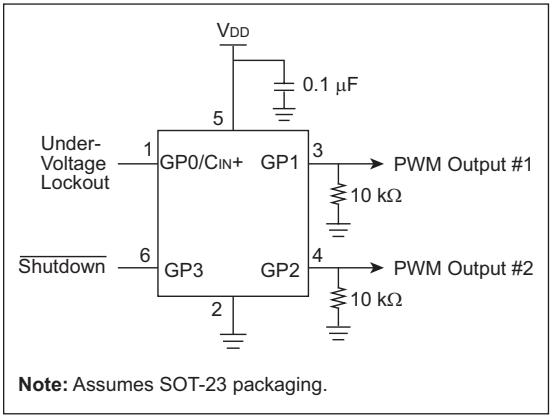
Application notes can be found on the Microchip web site at www.microchip.com.

TIP #2 A Start-Up Sequencer

Some new devices have multiple voltage requirements (e.g., core voltages, I/O voltages, etc.). The sequence in which these voltages rise and fall may be important.

By expanding on the previous tip, a start-up sequencer can be created to control two output voltages. Two PWM outputs are generated to control the shutdown pins of two SMPS controllers. Again, this type of control only works on controllers that respond quickly to changes on the shutdown pin (such as those that do cycle-by-cycle limiting).

Figure 2-1: Multiple PWM Output Soft-Start Controller



Note: Assumes SOT-23 packaging.

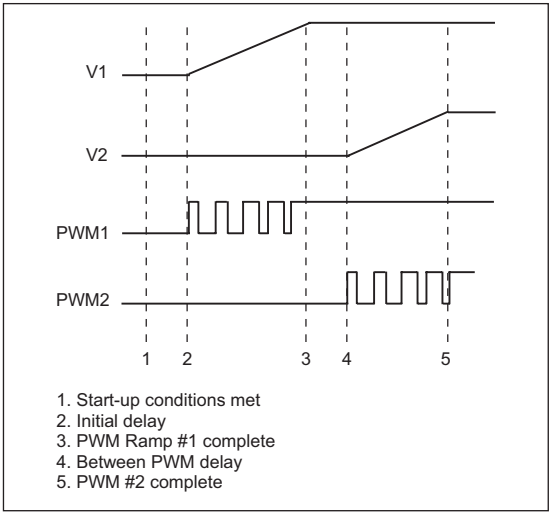
This design uses the PIC MCU comparator to implement an under-voltage lockout. The input on the GP0/CIN+ pin must be above the internal 0.6V reference for soft-start to begin, as shown in Figure 2-2.

Two conditions must be met in order for the soft-start sequence to begin:

1. The shutdown pin must be held at VDD (logic high).
2. The voltage on GP0 must be above 0.6V.

Once both start-up conditions are met, the sequences will delay and PWM #1 will ramp from 0% to 100%. A second delay allows the first voltage to stabilize before the sequencer ramps PWM #2 from 0% to 100%. All delays and ramp times are under software control and can be customized for specific applications. If either soft-start condition becomes invalid, the circuit will shutdown the SMPS controllers.

Figure 2-2: Timing Diagram



Example software is provided for the PIC10F200 which was taken from TB093, “*Multiple PWM Output Soft-Start Controller for Switching Power Supplies*” (DS91093).

TIP #3 A Tracking and Proportional Soft-Start of Two Power Supplies

Expanding on the previous tip, we can also use a PIC MCU to ensure that two voltages in a system rise together or rise proportionally to one another, as shown in Figure 3-1. This type of start-up is often used in applications with devices that require multiple voltages (such as I/O and core voltages).

Like the previous two, this tip is designed to control the shutdown pin of the SMPS controller and will only work with controllers that respond quickly to changes on the shutdown pin.

Figure 3-1: Timing Diagram

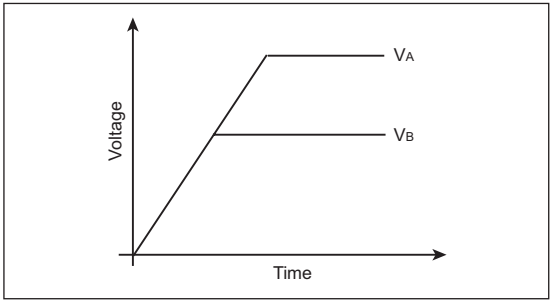
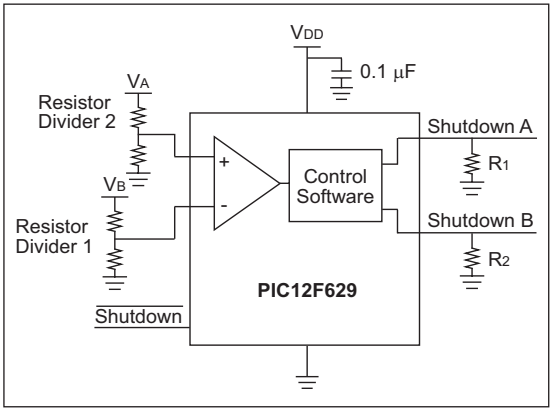


Figure 3-2: Example Schematic



The comparator of the PIC MCU is used to determine which voltage is higher and increases the on-time of the other output accordingly. The logic for the shutdown pins is as shown in Table 3-1.

Table 3-1: Shutdown Pin Logic

Case	Shutdown A	Shutdown B
$V_A > V_B$	Low	High
$V_B > V_A$	High	Low
V_B > Internal Reference	High	High

To determine if it has reached full voltage, V_B is compared to the internal voltage reference. If V_B is higher, both shutdown outputs are held high.

Resistor Divider 1 should be designed so that the potentiometer output is slightly higher than the comparator voltage reference when V_B is at full voltage.

The ratio of resistors in Resistor Divider 2 can be varied to change the slope at which V_A rises.

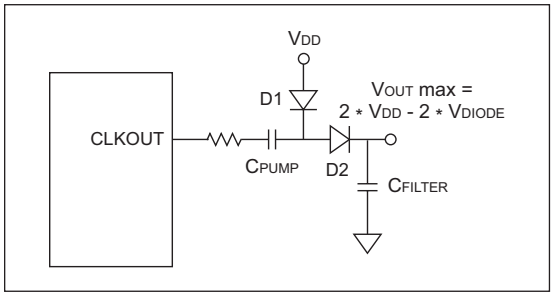
Pull-down resistors ensure the power supplies will not operate unexpectedly when the PIC MCU is being reset.

TIP #10 Driving High Side FETs

In applications where high side N channel FETs are to be driven, there are several means for generating an elevated driving voltage. One very simple method is to use a voltage doubling charge pump as shown in Figure 10-1.

Method 1

Figure 10-1: Typical Change Pump

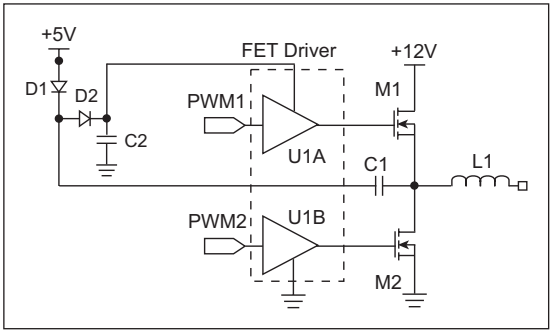


The PIC MCUs CLKOUT pin toggles at 1/4 of the oscillator frequency. When CLKOUT is low, D1 is forward biased and conducts current, thereby charging CPUMP. After CLKOUT is high, D2 is forward biased, moving the charge to CFILTER. The result is a voltage equal to twice the VDD minus two diode drops. This can be used with a PWM or any other I/O pin that toggles.

In Figure 10-2, a standard FET driver is used to drive both the high and low side FETs by using the diode and capacitor arrangement.

Method 2

Figure 10-2: Schematic

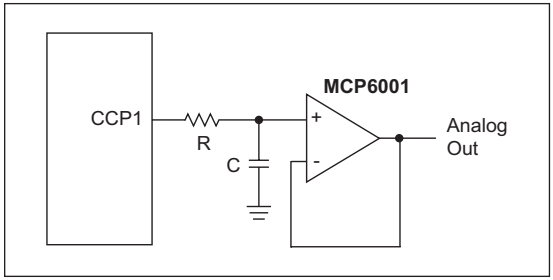


The +5V is used for powering the microcontroller. Using this arrangement, the FET driver would have approximately 12 + (5 - VDIODE) - VDIODE volts as a supply and is able to drive both the high and low side FETs.

The circuit above works by charging C1 through D1 to (5V - VDIODE) while M2 is on, effectively connecting C1 to ground. When M2 turns off and M1 turns on, one side of C1 is now at 12V and the other side is at 12V + (5V - VDIODE). The D2 turns on and the voltage supplied to the FET driver is 12V + (5V - VDIODE) - VDIODE.

TIP #11 Generating a Reference Voltage with a PWM Output

Figure 11-1: Low-Pass Filter



A PWM signal can be used to create a Digital-to-Analog Converter (DAC) with only a few external components. Conversion of PWM waveforms to analog signals involves the use of an analog low-pass filter. In order to eliminate unwanted harmonics caused by a PWM signal, the PWM frequency (F_{PWM}) should be significantly higher than the bandwidth (F_{BW}) of the desired analog signal. Equation 11-1 shows this relation.

Equation 11-1

$$F_{PWM} = K \bullet F_{BW}$$

Where harmonics decrease as K increases.

R and C are chosen based on the following equation:

Equation 11-2

$$RC = 1/(2 \bullet \pi \bullet F_{BW})$$

Where harmonics decrease as K increases.

Choose the R value based on drive capability and then calculate the required C value. The attenuation of the PWM frequency for a given RC filter is shown in Equation 11-3.

Equation 11-3

$$Att(dB) = -10 \bullet \log [1 + (2 \pi \bullet F_{PWM} \bullet RC)^2]$$

If the attenuation calculated in Equation 11-3 is not sufficient, then K must be increased in Equation 11-1.

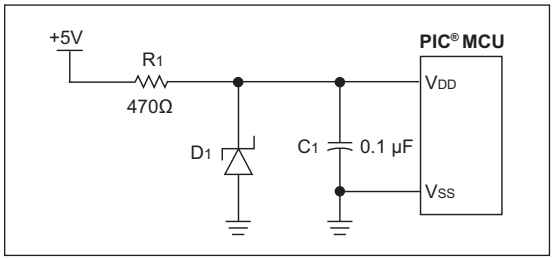
In order to sufficiently attenuate the harmonics, it may be necessary to use small capacitor values or large resistor values. Any current draw will effect the voltage across the capacitor. Adding an op amp allows the analog voltage to be buffered and, because of this, any current drawn will be supplied by the op amp and not the filter capacitor.

For more information on using a PWM signal to generate an analog output, refer to AN538, “*Using PWM to Generate Analog Output*” (DS00538).

TIP #2 Low-Cost Alternative Power System Using a Zener Diode

Details a low-cost regulator alternative using a Zener diode.

Figure 2-1: Zener Supply



A simple, low-cost 3.3V regulator can be made out of a Zener diode and a resistor as shown in Figure 2-1. In many applications, this circuit can be a cost-effective alternative to using a LDO regulator. However, this regulator is more load sensitive than a LDO regulator. Additionally, it is less energy efficient, as power is always being dissipated in R1 and D1.

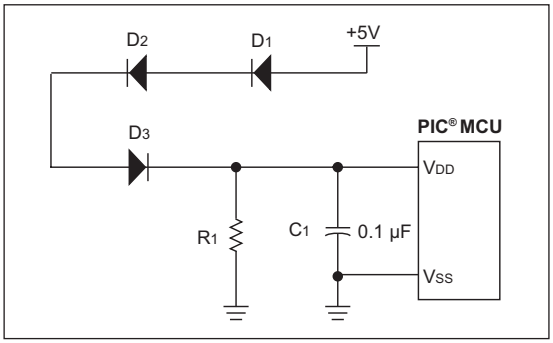
R1 limits the current to D1 and the PIC MCU so that V_{DD} stays within the allowable range. Because the reverse voltage across a Zener diode varies as the current through it changes, the value of R1 needs to be considered carefully.

R1 must be sized so that at maximum load, typically when the PIC MCU is running and is driving its outputs high, the voltage drop across R1 is low enough so that the PIC MCU has enough voltage to operate. Also, R1 must be sized so that at minimum load, typically when the PIC MCU is in Reset, that V_{DD} does not exceed either the Zener diode's power rating or the maximum V_{DD} for the PIC MCU.

TIP #3 Lower Cost Alternative Power System Using 3 Rectifier Diodes

Figure 3-1 details a lower cost regulator alternative using 3 rectifier diodes.

Figure 3-1: Diode Supply



We can also use the forward drop of a series of normal switching diodes to drop the voltage going into the PIC MCU. This can be even more cost-effective than the Zener diode regulator. The current draw from this design is typically less than a circuit using a Zener.

The number of diodes needed varies based on the forward voltage of the diode selected. The voltage drop across diodes D1-D3 is a function of the current through the diodes. R1 is present to keep the voltage at the PIC MCUs V_{DD} pin from exceeding the PIC MCUs maximum V_{DD} at minimum loads (typically when the PIC MCU is in Reset or sleeping). Depending on the other circuitry connected to V_{DD}, this resistor may have its value increased or possibly even eliminated entirely. Diodes D1-D3 must be selected so that at maximum load, typically when the PIC is running and is driving its outputs high, the voltage drop across D1-D3 is low enough to meet the PIC MCUs minimum V_{DD} requirements.

TIP #7 3.3V → 5V Using a Diode Offset

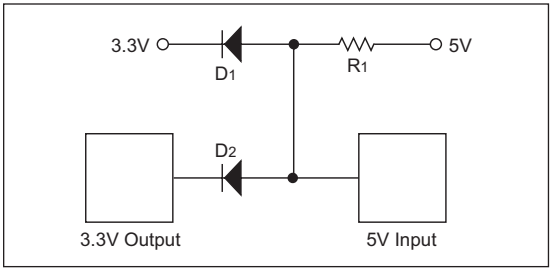
The inputs voltage thresholds for 5V CMOS and the output drive voltage for 3.3V LVTTL and LVCMOS are listed in Table 7-1.

Table 7-1: Input/Output Thresholds

	5V CMOS Input	3.3V LVTTL Output	3.3V LVCMOS Output
High Threshold	> 3.5V	> 2.4V	> 3.0V
Low Threshold	< 1.5V	< 0.4V	< 0.5V

Note that both the high and low threshold input voltages for the 5V CMOS inputs are about a volt higher than the 3.3V outputs. So, even if the output from the 3.3V system could be offset, there would be little or no margin for noise or component tolerance. What is needed is a circuit that offsets the outputs and increases the difference between the high and low output voltages.

Figure 7-1: Diode Offset



When output voltage specifications are determined, it is done assuming that the output is driving a load between the output and ground for the high output, and a load between 3.3V and the output for the low output. If the load for the high threshold is actually between the output and 3.3V, then the output voltage is actually much higher as the load resistor is the mechanism that is pulling the output up, instead of the output transistor.

If we create a diode offset circuit (see Figure 7-1), the output low voltage is increased by the forward voltage of the diode D1, typically 0.7V, creating a low voltage at the 5V CMOS input of 1.1V to 1.2V. This is well within the low threshold input voltage for the 5V CMOS input. The output high voltage is set by the pull-up resistor and diode D2, tied to the 3.3V supply. This puts the output high voltage at approximately 0.7V above the 3.3V supply, or 4.0 to 4.1V, which is well above the 3.5V threshold for the 5V CMOS input.

Note: For the circuit to work properly, the pull-up resistor must be significantly smaller than the input resistance of the 5V CMOS input, to prevent a reduction in the output voltage due to a resistor divider effect at the input. The pull-up resistor must also be large enough to keep the output current loading on the 3.3V output within the specification of the device.

Neglecting the affects of R_S and R_L, the formula for determining the values for R1 and R2 is given by Equation 12-1.

Equation 12-1: Divider Values

$$\frac{V_S}{R_1 + R_2} = \frac{V_L}{R_2} \quad \text{; General relationship}$$

$$R_1 = \frac{(V_S - V_L) \cdot R_2}{V_L} \quad \text{; Solving for R1}$$

$$R_1 = 0.515 \cdot R_2 \quad \text{; Substituting voltages}$$

The formula for determining the rise and fall times is given in Equation 12-2. For circuit analysis, the Thevenin equivalent is used to determine the applied voltage, V_A, and the series resistance, R. The Thevenin equivalent is defined as the open circuit voltage divided by the short circuit current. The Thevenin equivalent, R, is determined to be 0.66*R1 and the Thevenin equivalent, V_A, is determined to be 0.66*V_S for the circuit shown in Figure 12-2 according to the limitations imposed by Equation 12-2.

Equation 12-2: Rise/Fall Time

$$t = -\left[R \cdot C \cdot \ln\left(\frac{V_F - V_A}{V_I - V_A}\right)\right]$$

Where:

- t = Rise or Fall time
- R = 0.66*R_1
- C = C_S+C_L
- V_I = Initial voltage on C (V_L)
- V_F = Final voltage on C (V_L)
- V_A = Applied voltage (0.66*V_S)

As an example, suppose the following conditions exist:

- Stray capacitance = 30 pF
- Load capacitance = 5 pF
- Maximum rise time from 0.3V to 3V ≤ 1 µS
- Applied source voltage Vs = 5V

The calculation to determine the *maximum* resistances is shown in Equation 12-3.

Equation 12-3: Example Calculation

Solve Equation 12-2 for *R*:

$$R = -\left[\frac{t}{C \cdot \ln\left(\frac{V_F - V_A}{V_I - V_A}\right)}\right]$$

Substitute values:

$$R = -\left[\frac{10 \cdot 10^{-7}}{35 \cdot 10^{-12} \cdot \ln\left(\frac{3 - (0.66 \cdot 5)}{0.3 - (0.66 \cdot 5)}\right)}\right]$$

Thevenin equivalent maximum R:

$R = 12408$

Solve for maximum R1 and R2:

$R1 = 0.66 \cdot R$ $\qquad$ $R2 = \frac{R1}{0.515}$

$R1 = 8190$ $\qquad$ $R2 = 15902$

TIP #16 5V → 3.3V Active Analog Attenuator

Reducing a signal's amplitude from a 5V to 3.3V system using an op amp.

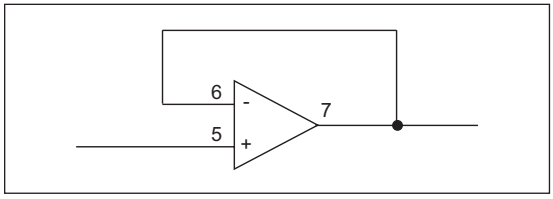
The simplest method of converting a 5V analog signal to a 3.3V analog signal is to use a resistor divider with a ratio R1:R2 of 1.7:3.3. However, there are a few problems with this.

1. The attenuator may be feeding a capacitive load, creating an unintentional low pass filter.
2. The attenuator circuit may need to drive a low-impedance load from a high-impedance source.

Under either of these conditions, an op amp becomes necessary to buffer the signals.

The op amp circuit necessary is a unity gain follower (see Figure 16-1).

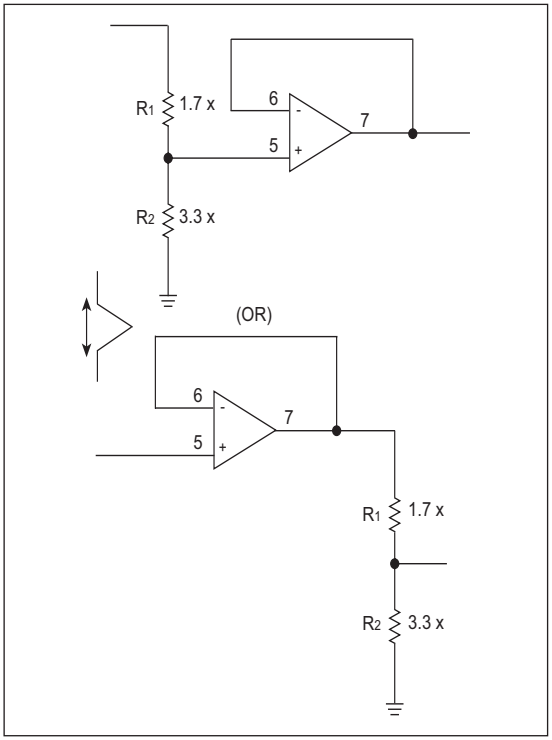
Figure 16-1: Unity Gain



This circuit will output the same voltage that is applied to the input.

To convert the 5V signal down to a 3V signal, we simply add the resistor attenuator.

Figure 16-2: Op Amp Attenuators



If the resistor divider is before the unity gain follower, then the lowest possible impedance is provided for the 3.3V circuits. Also, the op amp can be powered from 3.3V, saving some power. If the X is made very large, then power consumed by the 5V side can be minimized.

If the attenuator is added after the unity gain follower, then the highest possible impedance is presented to the 5V source. The op amp must be powered from 5V and the impedance at the 3V side will depend upon the value of R1||R2.