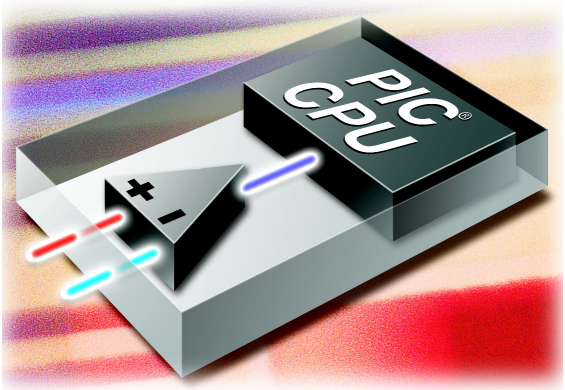


MICROCHIP

PIC® MCU
Comparator
Tips ‘n Tricks

Table of Contents

Tips ‘n Tricks Introduction

Tips ‘n Tricks

NOTES:

TIPS ‘N TRICKS INTRODUCTION

Microchip continues to provide innovative products that are smaller, faster, easier to use and more reliable. The Flash-based PIC® microcontrollers (MCU) are used in a wide range of everyday products from smoke detectors to industrial, automotive and medical products.

The PIC12F/16F Family of devices with on-chip voltage comparators merge all the advantages of the PIC MCU architecture and the flexibility of Flash program memory with the mixed signal nature of a voltage comparator. Together they form a low-cost hybrid digital/analog building block with the power and flexibility to work in an analog world.

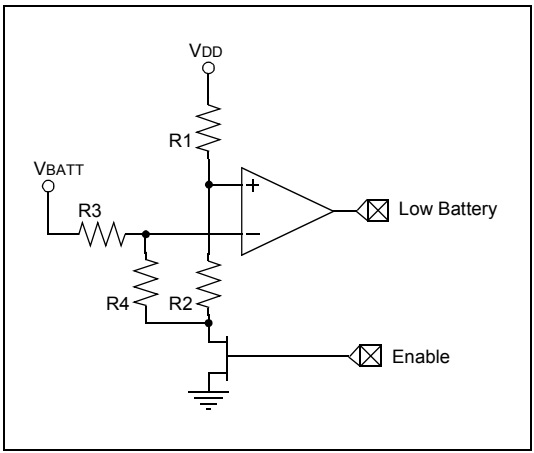
The flexibility of Flash and an excellent development tool suite, including a low-cost In-Circuit Debugger, In-Circuit Serial Programming™ (ICSP™) and MPLAB® ICE 2000 emulation, make these devices ideal for just about any embedded control application.

The following series of Tips ‘n Tricks can be applied to a variety of applications to help make the most of discrete voltage comparators or microcontrollers with on-chip voltage comparators.

TIP #1 Low Battery Detection

When operating from a battery power supply, it is important for a circuit to be able to determine when the battery charge is insufficient for normal operation of the circuit. Typically, this is a comparator-based circuit similar to the Programmable Low Voltage Detect (PLVD) peripheral. If the PLVD peripheral is not available in the microcontroller, a similar circuit can be constructed using a comparator and a few external components (see Figure 1-1 and Figure 1-2). The circuit in Figure 1-1 assumes that the microcontroller is operating from a regulated supply voltage. The circuit in Figure 1-2 assumes that the microcontroller supply is unregulated.

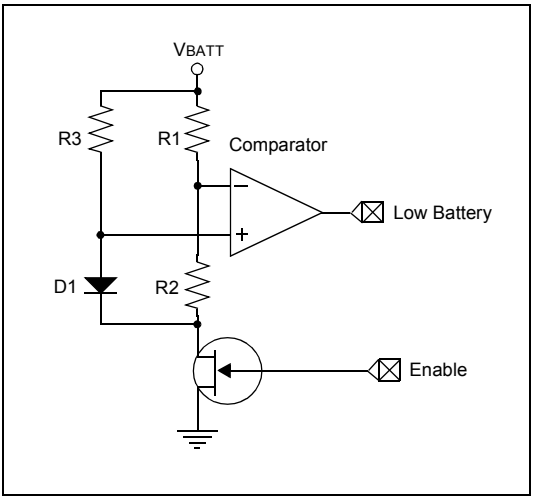
FIGURE 1-1: REGULATED SUPPLY



The comparator will trip when the battery voltage, V_{BATT} = 5.7V: R1 = 33k, R2 = 10k, R3 = 39k, R4 = 10k, V_{DD} = 5V.

In Figure 1-1, resistors R1 and R2 are chosen to place the voltage at the non-inverting input at approximately 25% of V_{DD}. R3 and R4 are chosen to set the inverting input voltage equal to the non-inverting input voltage when the battery voltage is equal to the minimum operating voltage for the system.

FIGURE 1-2: UNREGULATED SUPPLY



Comparator will trip when VBATT = 3V: R1 = 33k, R2 = 10k and R3 = 470Ω.

In Figure 1-2, resistor R3 is chosen to bias diode D1 above its forward voltage when VBATT is equal to the minimum battery voltage for the system. Resistors R1 and R2 are chosen to set the inverting input voltage equal to the forward voltage of D1.

TIP #2 Faster Code for Detecting Change

When using a comparator to monitor a sensor, it is often just as important to know when a change occurs as it is to know what the change is. To detect a change in the output of a comparator, the traditional method has been to store a copy of the output and periodically compare the held value to the actual output to determine the change. An example of this type of routine is shown below.

EXAMPLE 2-1:

```
Test
  MOVF  hold,w   ; get old Cout
  XORWF CMCON,w  ; compare to new Cout
  ANDLW COUTMASK
  BTFSC STATUS,Z
  RETLW 0        ; if = return "no change"
  MOVF  CMCON,w  ; if not =, get new Cout
  ANDLW COUTMASK ; remove all other bits
  MOVWF hold     ; store in holding var.
  IORLW CHNGBIT  ; add change flag
  RETURN
```

This routine requires 5 instructions for each test, 9 instructions if a change occurs, and 1 RAM location for storage of the old output state.

A faster method for microcontrollers with a single comparator is to use the comparator interrupt flag to determine when a change has occurred.

EXAMPLE 2-2:

```
Test
  BTFSS PIR1,CMIF  ;test comparator flag
  RETLW 0          ;if clear, return a 0
  BTFSS CMCON,COUT ;test Cout
  RETLW CHNGBIT    ;if clear return
                   ;CHNGFLAG
  RETLW COUTMASK + CHNGBIT;if set,
                   ;return both
```

This routine requires 2 instructions for each test, 3 instructions if a change occurs, and no RAM storage.

If the interrupt flag can not be used, or if two comparators share an interrupt flag, an alternate method that uses the comparator output polarity bit can be used.

EXAMPLE 2-3:

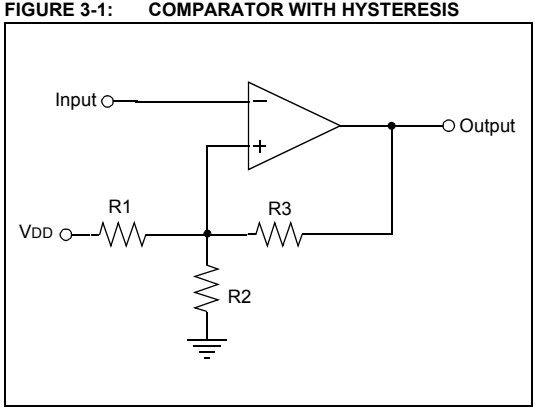
```
Test
  BTFSS CMCON,COUT ;test Cout
  RETLW 0          ;if clear, return 0
  MOVLW CINVBIT    ;if set, invert Cout
  XORWF CMCON,f    ;forces Cout to 0
  BTFSS CMCON,CINV ;test Cout polarity
  RETLW CHNGFLAG   ;if clear, return
                   ;CHNGFLAG
  RETLW COUTMASK + CHNGFLAG;if set,
                   ;return both
```

This routine requires 2 instructions for each test, 5 instructions if a change occurs, and no GPR storage.

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 megohm, 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

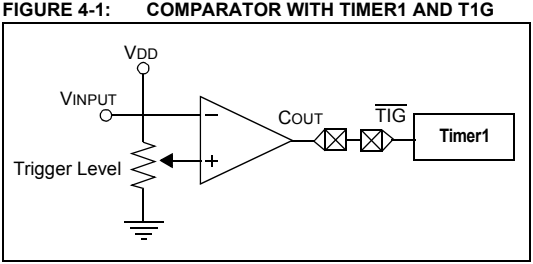
TIP #4 Pulse Width Measurement

To measure the high or low pulse width of an incoming analog signal, the comparator can be combined with Timer1 and the Timer1 Gate input option (see Figure 4-1). Timer1 Gate acts as a count enable for Timer1. If the input is low, Timer1 will count. If the $\overline{T1G}$ input is high, Timer1 does not count. Combining $\overline{T1G}$ with the comparator allows the designer to measure the time between a high-to-low output change and a low-to-high output change.

To make a measurement between a low-to-high and a high-to-low transition, the only change required is to set the CINV bit in the comparator CMCON register which inverts the comparator output.

Because the output of the comparator can change asynchronously with the Timer1 clock, only comparators with the ability to synchronize their output with the Timer1 clock should be used and their C2SYNC bits should be set.

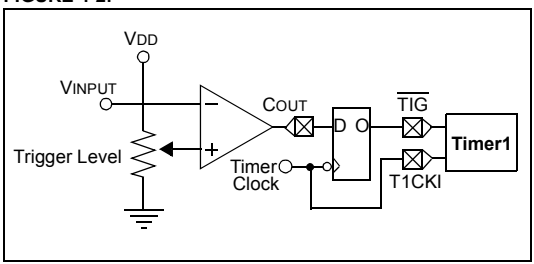
FIGURE 4-1: COMPARATOR WITH TIMER1 AND $\overline{T1G}$



If the on-chip comparator does not have the ability to synchronize its output to the Timer1 clock, the output can be synchronized externally using a discrete D flip-flop (see Figure 4-2).

Note: The flip-flop must be falling edge triggered to prevent a race condition.

FIGURE 4-2:

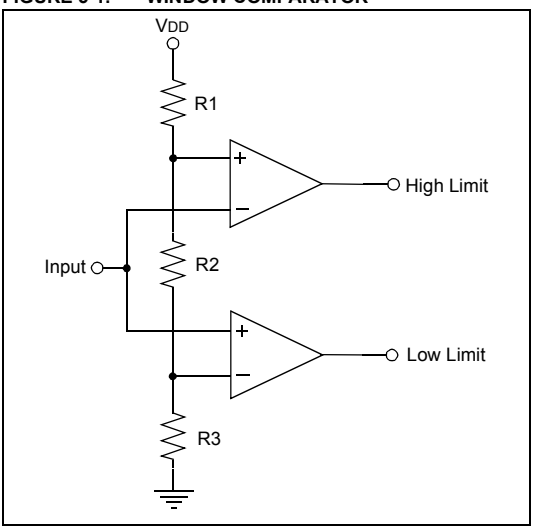


TIP #5 Window Comparison

When monitoring an external sensor, it is often convenient to be able to determine when the signal has moved outside a pre-established safe operating range of values or window of operation. This windowing provides the circuit with an alarm when the signal moves above or below safety limits, ignoring minor fluctuations inside the safe operating range.

To implement a window comparator, two voltage comparators and 3 resistors are required (see Figure 5-1).

FIGURE 5-1: WINDOW COMPARATOR



Resistors R1, R2 and R3 form a voltage divider which generates the high and low threshold voltages. The outputs HIGH LIMIT and LOW LIMIT are both active high, generating a logic one on the HIGH LIMIT output when the input voltage rises above the high threshold, and a logic one on the LOW LIMIT output when the input voltage falls below the low threshold.

To calculate values for R1, R2 and R3, find values that satisfy Equation 5-1 and Equation 5-2.

Note: A continuous current will flow through R1, R2 and R3. To limit the power dissipation in the resistors, the total resistance of R1, R2 and R3 should be at least 1k. The total resistance of R1, R2 and R3 should also be kept less than 1 megohm to prevent offset voltages due to the input bias currents of the comparator.

EQUATION 5-1:

$$V_{TH\text{-}HI} = \frac{V_{DD} * (R_3 + R_2)}{R1 + R2 + R3}$$

EQUATION 5-2:

$$V_{TH\text{-}LO} = \frac{V_{DD} * R_3}{R1 + R2 + R3}$$

Example:

- V_{DD} = 5.0V, V_{TH} = 2.5V, V_{TL} = 2.0V
- R_1 = 12k, R_2 = 2.7k, R_3 = 10k
- V_{TH} (actual) = 2.57V, V_{TL} (actual) = 2.02V

Adding Hysteresis:

To add hysteresis to the HIGH LIMIT comparator, follow the procedure outlined in TIP #3. Use the series combination of R2 and R3 as the resistor R2 in TIP #3.

To add hysteresis to the LOW LIMIT comparator, choose a suitable value for Req, 1k to 10 kOhm, and place it between the circuit input and the non-inverting input of the LOW LIMIT comparator. Then calculate the needed feedback resistor using Equation 3-4 and Equation 3-5.

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.

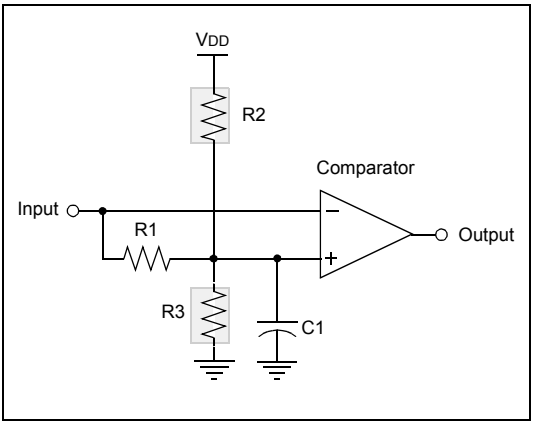
The data slicer shown in Figure 6-1 operates by comparing the incoming signal with a sliding reference derived from the average DC value of the incoming signal. The DC average value is found using a simple RC low-pass filter (R1 and C1). The corner frequency of the RC filter should be high enough to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

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.

FIGURE 6-1: DATA SLICER



TIP #7 One-Shot

When dealing with short duration signals or glitches, it is often convenient to stretch out the event using a mono-stable, multi-vibrator or one-shot. Whenever the input pulses, the one-shot fires holding its output for a preset period of time. This stretches the short trigger input into a long output which the microcontroller can capture.

The circuit is designed with two feedback paths around a comparator. The first is a positive hysteresis feedback which sets a two level threshold, V_{HI} and V_{LO}, based on the state of the comparator output. The second feedback path is an RC time circuit.

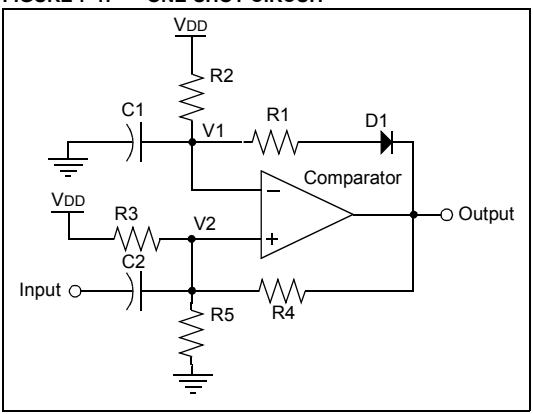
The one-shot circuit presented in Figure 7-1 is triggered by a low-high transition on its input and generates a high output pulse. Using the component values from the example, the circuit’s operation is as follows.

Prior to triggering, C1 will have charged to a voltage slightly above 0.7V due to resistor R2 and D1 (R1 << R2 and will have only a minimal effect on the voltage). The comparator output will be low, holding the non-inverting input slightly below 0.7V due to the hysteresis feedback through R3, R4 and R5 (the hysteresis lower limit is designed to be less than 0.7V). With the non-inverting input held low, C2 will charge up to the difference between the circuit input and the voltage present at the non-inverting input.

When the circuit input is pulsed high, the voltage present at the non-inverting input is pulled above 0.7V due to the charge in C2. This causes the output of the comparator to go high, the hysteresis voltage at the non-inverting input goes to the high threshold voltage, and C1 begins charging through R2.

When the voltage across C1 exceeds the high threshold voltage, the output of the comparator goes low, C1 is discharged to just above the 0.7V limit, the non-inverting input is pulled below 0.7V, and the circuit is reset for the next pulse input, waiting for the next trigger input.

FIGURE 7-1: ONE-SHOT CIRCUIT



To design the one-shot, first create the hysteresis feedback using the techniques from TIP #3. Remember to set the low threshold below 0.7V. Next, choose values for R2 and C1 using Equation 7-1.

EQUATION 7-1:

$$T_{PULSE} = \frac{R2 * C1 * \ln(V_{TH}/V_{TL})}{4}$$

D1 can be any low voltage switching diode. R1 should be 1% to 2% of R2 and C2 should be between 100 and 220 pF.

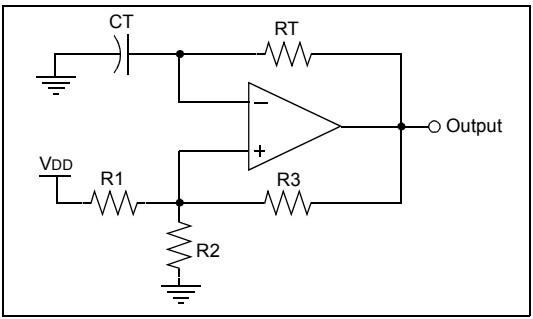
Example:

- V_{DD} = 5V, V_{TH} = 3.0V, V_{TL} = 2.5V
- From TIP #3, R4 = 1k, R5 = 1.5k and R3 = 12k
- T_{PULSE} = IMS, C1 = .1 µF and R2 = 15k
- D1 is a 1N4148, R1 = 220Ω and C2 = 150 pF

TIP #8 Multi-Vibrator (Square Wave Output)

A multi-vibrator is an oscillator designed around a voltage comparator or operational amplifier (see Figure 8-1). Resistors R1 through R3 form a hysteresis feedback path from the output to the non-inverting input. Resistor RT 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 RT. When the voltage across CT reaches the low threshold voltage, the output is forced high and the cycle starts over.

FIGURE 8-1: MULTI-VIBRATOR CIRCUIT



To design a multi-vibrator, first design the hysteresis feedback path using the procedure in TIP #3. Be careful to choose threshold voltages (V_{TH} and V_{TL}) that are evenly spaced within the common mode range of the comparator and centered on $V_{DD}/2$. 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 8-1 defines the relationship between RT, CT, V_{TH}, V_{TL} and F_{OSC}.

EQUATION 8-1:

$$F_{OSC} = \frac{1}{2 * RT * CT * \ln(V_{TH}/V_{TL})}$$

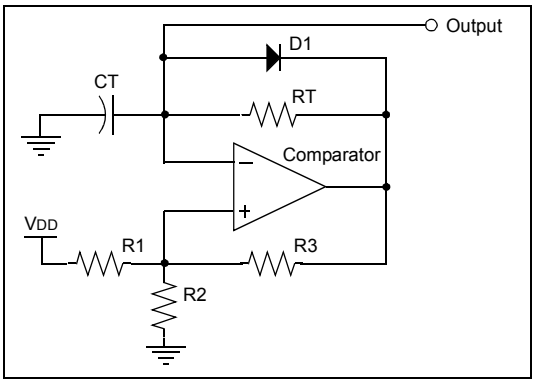
Example:

- V_{DD} = 5V, V_{TH} = 3.333, V_{TL} = 1.666V
- R1, to R2, to R3 = 10k
- R_T = 15 kHz, C_T = .1 µF for F_{OSC} = 480 Hz

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:

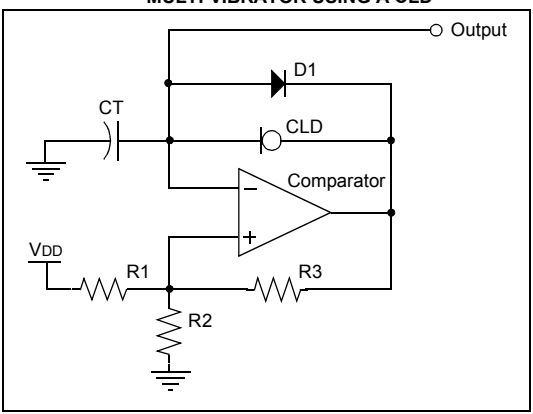
- V_{DD} = 5V, V_{TL} = 1.666V and V_{TH} = 3.333V
- R1, R2 and R3 = 10k
- R_T = 15k, C_T = .1 µF for a F_{OSC} = 906 Hz

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:

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

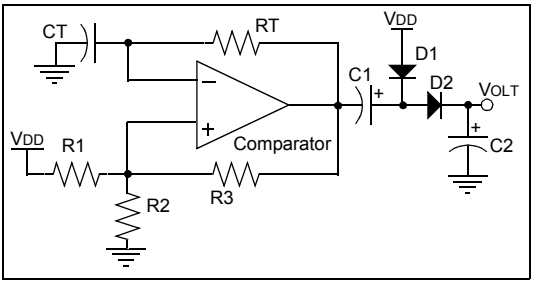
FIGURE 9-2: ALTERNATE RAMP WAVEFORM MULTI-VIBRATOR USING A CLD



TIP #10 Capacitive Voltage Doubler

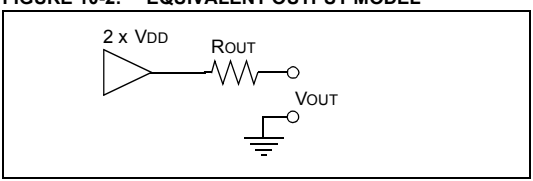
This tip takes the multi-vibrator described in TIP #8 and builds a capacitive voltage doubler around it (see Figure 10-1). The circuit works by alternately charging capacitor C1 through diode D1, and then charge balancing the energy in C1 with C2 through diode D2. At the start of the cycle, the output of the multi-vibrator is low and charge current flows from V_{DD} through D1 and into C1. When the output of the multi-vibrator goes high, D1 is reverse biased and the charge current stops. The voltage across C1 is added to the output voltage of the multi-vibrator, creating a voltage at the positive terminal of C1 which is 2 x V_{DD}. This voltage forward biases D2 and the charge in C1 is shared with C2. When the output of the multi-vibrator goes low again, the cycle starts over.

FIGURE 10-1: CAPACITIVE VOLTAGE DOUBLER



Note: The output voltage of a capacitive double is unregulated and will sag with increasing load current. Typically, the output is modeled as a voltage source with a series resistance (see Figure 10-2).

FIGURE 10-2: EQUIVALENT OUTPUT MODEL



To design a voltage doubler, first determine the maximum tolerable output resistance based on the required output current and the minimum tolerable output voltage. Remember that the output current will be limited to one half of the output capability of the comparator. Then choose a transfer capacitance and switching frequency using Equation 10-1.

EQUATION 10-1:

$$R_{OUT} = \frac{1}{F_{SWITCH} * C1}$$

Note: R_{OUT} will be slightly higher due to the dynamic resistance of the diodes. The equivalent series resistance or ESR, of the capacitors and the output resistance of the comparator. See the data sheet for the TC7660 for a more complete description.

Once the switching frequency is determined, design a square-wave multi-vibrator as described in TIP #8.

Finally, select diodes D1 and D2 for their current rating and set C2 equal to C1.

Example:

From TIP #8, the values are modified for a F_{OSC} of 4.8 kHz.

- C1 and C2 = 10 μF
- R_{OUT} = 21Ω.

TIP #11 PWM Generator

This tip shows how the multi-vibrator (ramp wave) can be used to generate a voltage controlled PWM signal. The ramp wave multi-vibrator operates as described in TIP #9, generating a positive going ramp wave. A second comparator compares the instantaneous voltage of the ramp wave with the incoming voltage to generate the PWM output (see Figure 11-2).

When the ramp starts, it is below the input voltage, and the output of the second comparator is pulled high starting the PWM pulse. The output remains high until the ramp wave voltage exceeds the input, then the output of the second comparator goes low ending the PWM pulse. The output of the second comparator remains low for the remainder of the ramp waveform. When the ramp waveform returns to zero at the start of the next cycle, the second comparator output goes high again and the cycle starts over.

FIGURE 11-1: PWM WAVE FORMS

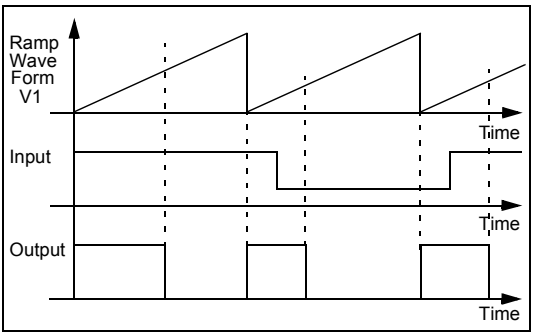
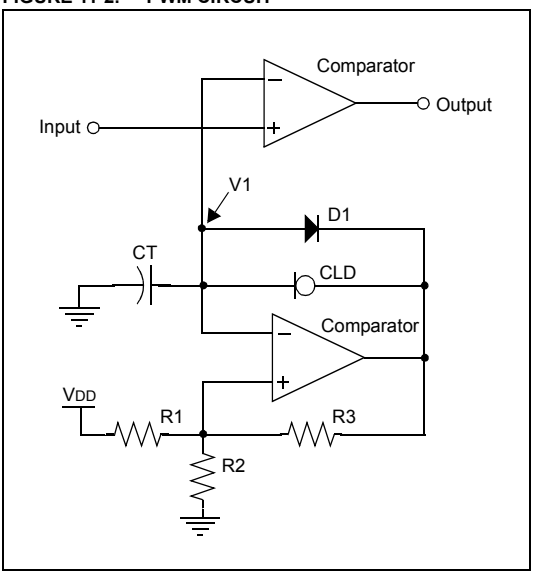


FIGURE 11-2: PWM CIRCUIT



To design a PWM generator, start with the design of a ramp wave multi-vibrator using the design procedure from TIP #9. Choose high and low threshold voltages for the multi-vibrators hysteresis feedback that are slightly above and below the desired PWM control voltages.

Note: The PWM control voltage will produce a 0% duty cycle for inputs below the low threshold of the multi-vibrator. A control voltage greater than the high threshold voltage will produce a 100% duty cycle output.

Using the example values from TIP #9 will result in a minimum pulse width at an input voltage of 1.7V and a maximum at an input of 3.2V.

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

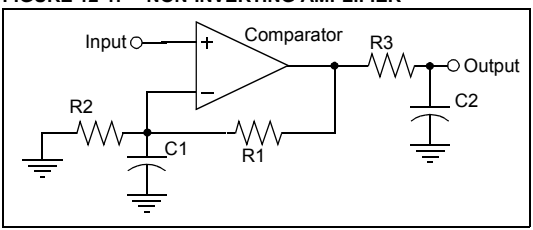
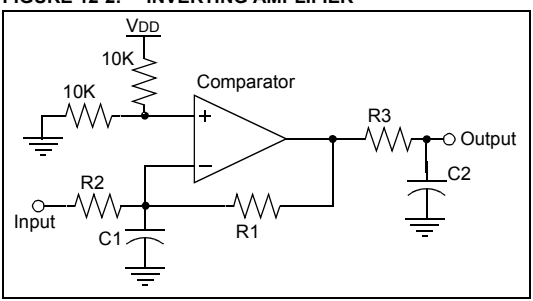


FIGURE 12-2: INVERTING AMPLIFIER



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:

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

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:

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

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:

$$Gain = \frac{R_1}{R_2}$$

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:

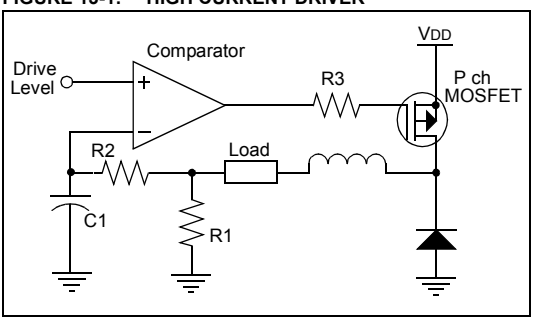
- For C_2 will set the corner F
- Gain = 6.156, R1 = R3 = 19.8k
- R2 = 3.84k, C1 = .047 μF, F_{CORNER} = 171 Hz
- C2 = .22 μF

TIP #13 PWM High-Current Driver

This tip combines a comparator with a MOSFET transistor and an inductor to create a switch mode high-current driver circuit. (See Figure 13-1).

The operation of the circuit begins with the MOSFET off and no current flowing in the inductor and load. With the sense voltage across R1 equal to zero and a DC voltage present at the drive level input, the output of the comparator goes low. The low output turns on the MOSFET and a ramping current builds through the MOSFET, inductor, load and R1.

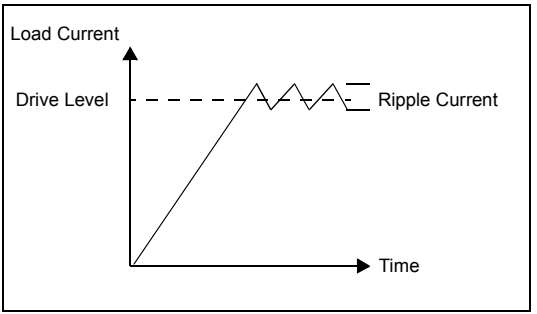
FIGURE 13-1: HIGH CURRENT DRIVER



When the current ramps high enough to generate a voltage across R1 equal to the drive level, the comparator output goes high turning off the MOSFET. The voltage at the junction of the MOSFET and the inductor then drops until D1 forward biases. The current continues ramping down from its peak level toward zero. When the voltage across the sense resistor R1 drops below the drive level, the comparator output goes low, the MOSFET turns on, and the cycle starts over.

R2 and C1 form a time delay network that limits the switching speed of the driver and causes it to slightly overshoot and undershoot the drive level when operating. The limit is necessary to keep the switching speed low, so the MOSFET switches efficiently. If R2 and C1 were not present, the system would run at a speed set by the comparator propagation delay and the switching speed of the MOSFET. At that speed, the switching time of the MOSFET would be a significant portion of the switching time and the switching efficiency of the MOSFET would be too low.

FIGURE 13-2: CURRENT THROUGH THE LOAD



To design a PWM high current driver, first determine a switching speed (FSWX) that is appropriate for the system. Next, choose a MOSFET and D1 capable of handling the load current requirements. Then choose values for R2 and C1 using Equation 13-1.

EQUATION 13-1:

$$F_{SWX} = \frac{2}{R2 * C1}$$

Next determine the maximum ripple current that the load will tolerate, and calculate the required inductance value for L1 using Equation 13-2.

EQUATION 13-2:

$$L = \frac{V_{DD} - V_{LOAD}}{I_{RIPPLE} * F_{SWX} * 2}$$

Finally, choose a value for R1 that will produce a feedback ripple voltage of 100 mV for the maximum ripple current I_{RIPPLE}.

Example:

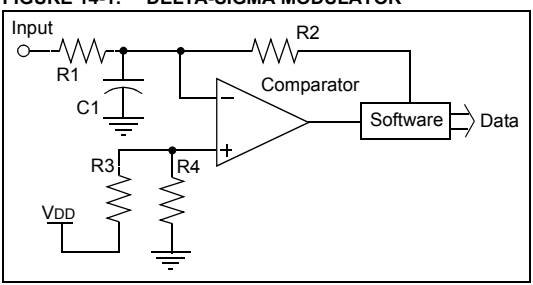
- F_{SWX} = 10 kHz, R2 = 22k, C1 = .01 μF
- I_{RIPPLE} = 100 mA, V_{DD} = 12V, V_L = 3.5V
- L = 4.25 mH

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 VDD/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:

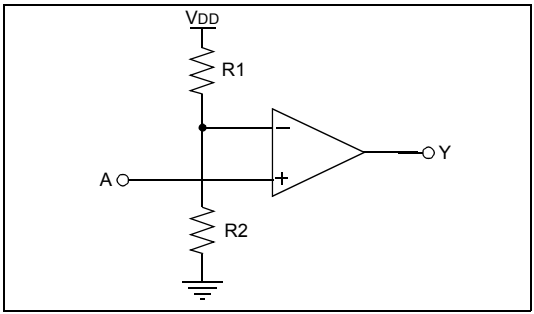
- R3 = R4 = 10 kHz
- R1 = R2 = 5.1k
- C1 = 1000 pF

TIP #15 Level Shifter

This tip shows the use of the comparator as a digital logic level shifter. The inverting input is biased to the center of the input voltage range ($V_{IN}/2$). The non-inverting input is then used for the circuit input. When the input is below the $V_{IN}/2$ threshold, the output is low. When the input is above $V_{IN}/2$, then the output is high. Values for R1 and R2 are not critical, though their ratio should result in a threshold voltage $V_{IN}/2$ at the mid-point of the input signal voltage range. Some microcontrollers have the option to connect the inverting input to an internal voltage reference. To use the reference in place of R1 and R2, simply select the internal reference and configure it for one half the input voltage range.

Note: Typical propagation delay for the circuit is 250-350 ns using the typical on-chip comparator peripheral of a microcontroller.

FIGURE 15-1: LEVEL SHIFTER



Example:

- V_{IN} = 0 - 2V, $V_{IN}/2$ = 1V, V_{DD} = 5V
- R2 = 10k, R3 = 3.9k

TIP #16 Logic: Inverter

When designing embedded control applications, there is often the need for an external gate. Using the comparator, several simple gates can be implemented. This tip shows the use of the comparator as an inverter.

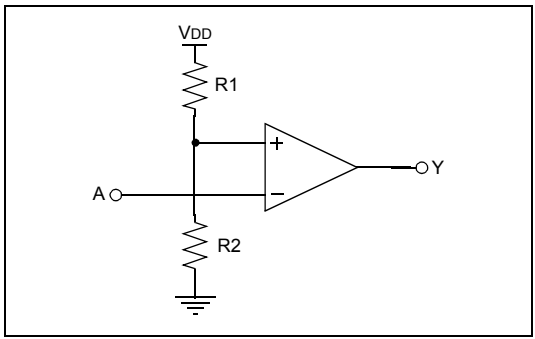
The non-inverting input is biased to the center of the input voltage range, typically $V_{DD}/2$. The inverting input is then used for the circuit input. When the input is below $V_{DD}/2$, the output is high. When the input is above $V_{DD}/2$, then the output is low.

Values for R1 and R2 are not critical, though they must be equal to set the threshold at $V_{DD}/2$.

Some microcontrollers have the option to connect the inverting input to an internal voltage reference. To use the reference in place of R1 and R2, move the input to the non-inverting input and set the output polarity bit in the comparator control register to invert the comparator output.

Note: Typical propagation delay for the circuit is 250-350 ns using the typical on-chip comparator peripheral of a microcontroller.

FIGURE 16-1: INVERTER



TIP #17 Logic: AND/NAND Gate

This tip shows the use of the comparator to implement an AND gate and its complement the NAND gate (see Figure 17-2). Resistors R1 and R2 drive the non-inverting input with 2/3 the supply voltage. Resistors R3 and R4 average the voltage of input A and B at the inverting input. If either A or B is low, the average voltage will be one half V_{DD} and the output of the comparator remains low. The output will go high only if both inputs A and B are high, which raises the input to the inverting input above 2/3 V_{DD}.

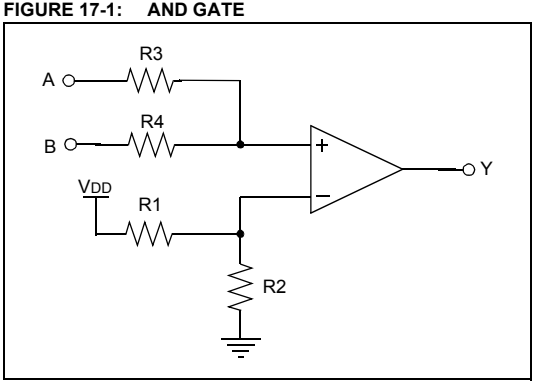
The operation of the NAND gate is identical to the AND gate, except that the output is inverted due to the swap of the inverting and non-inverting inputs.

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. The combination of R1 and R2 will draw current constantly, so they must be kept large to minimize current draw.
3. All resistances on the inverting input 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 R3 and R4.
4. Resistor R2 must be 2 x R1.
5. Resistor R3 must be equal to R4.

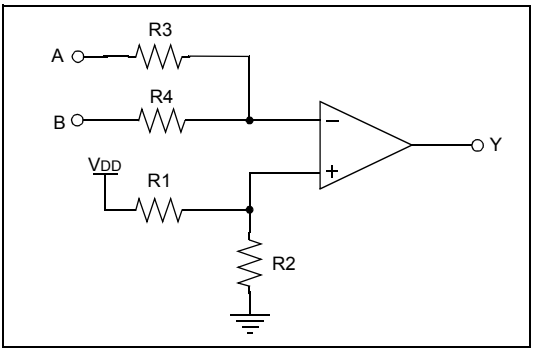
FIGURE 17-1: AND GATE



Example:

- V_{DD} = 5V, R3 = R4 = 10k
- R_1 = 5.1k, R2 = 10k

FIGURE 17-2: NAND GATE



TIP #18 Logic: OR/NOR Gate

This tip shows the use of the comparator to implement an OR gate, and its complement, the NOR gate.

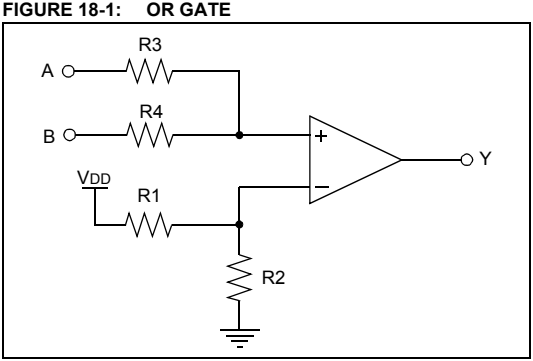
Resistors R1 and R2 drive the non-inverting input of the comparator with 1/3 V_{DD}. Resistors R3 and R4 average the voltages of the inputs A and B at the inverting input. If either A or B is high, the average voltage is 1/2 V_{DD} and the output of the comparator is high. Only if both A and B are low does the average voltage at the non-inverting input drop below 1/3 the supply voltage, causing the comparator output to go low. The operation of the NOR gate is identical to the OR gate, except the output is inverted due to the swap of the inverting and non-inverting inputs.

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 VDD for the circuit to operate properly.
2. The combination of R1 and R2 will draw current constantly, so they must be kept large to minimize current draw.
3. All resistances on the inverting input 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 R3 and R4.
4. Resistor R1 must be 2 x R2.
5. Resistor R3 must be equal to R4.

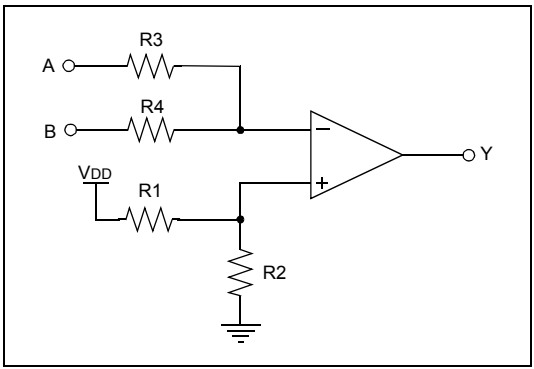
FIGURE 18-1: OR GATE



Example:

- V_{DD} = 5V, $R_3 = R_4$ = 10k
- R_1 = 10k, R_2 = 5.1k

FIGURE 18-2: NOR GATE



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 GND). 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 GND). 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

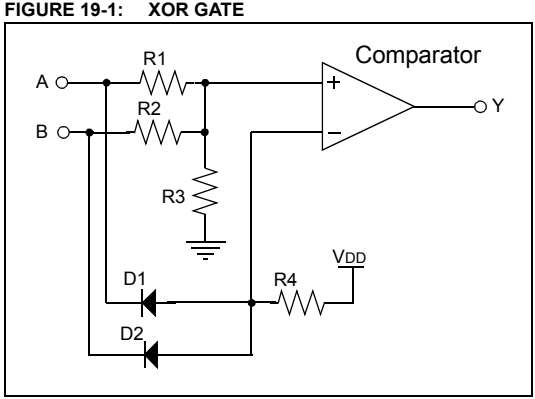
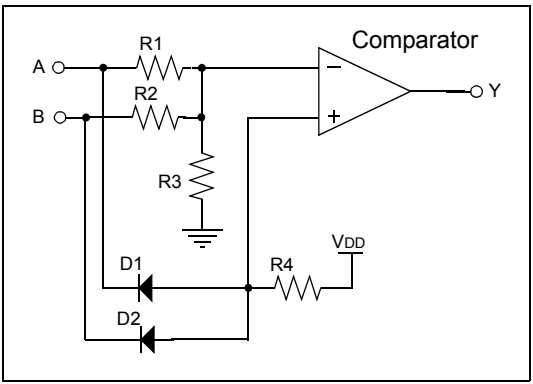


FIGURE 19-2: XNOR GATE



Example:

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

TIP #20 Logic: Set/Reset Flip Flop

This tip shows the use of the comparator to implement a Set/Reset Flip Flop.

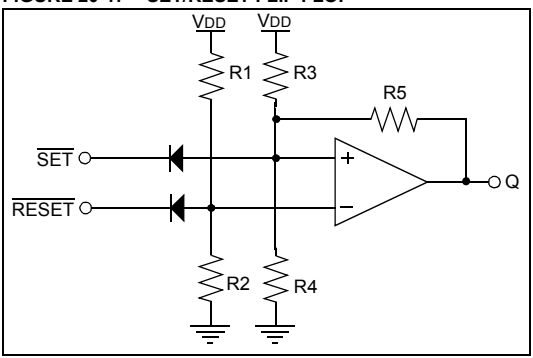
The inverting and non-inverting inputs are biases at $V_{DD}/2$ by resistors R1 through R4. The non-inverting input also receives positive feedback from the output through R5. The common bias voltages and the positive feedback configure the comparator as a bistable latch. If the output Q is high, the non-inverting input is also pulled high, which reinforces the high output. If Q is low, the non-inverting input is also pulled low, which reinforces the low output. To change state, the appropriate input must be pulled low to overcome the positive feedback. The diodes prevent a positive state on either input from pulling the bias of either input above $V_{DD}/2$.

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 Set and Reset must be driven near ground for the circuit to operate properly.
2. The combination of R1/R2 and R3/R4 will draw current constantly, so they must be kept large to minimize current draw.
3. R1 through R4 must be equal for a $V_{DD}/2$ trip level.
4. R5 must be greater or equal to R3.
5. R1 through R4 will react with the input capacitance of the comparator, so larger values will limit the minimum input pulse width.

FIGURE 20-1: SET/RESET FLIP FLOP



Example:

- Diodes = 1N4148
- R1 = R2 = R3 = R4 = 10k
- R_5 = 10k

NOTES:

NOTES:

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. Microchip makes no representations or warranties of any kind whether express or implied, written or oral, statutory or otherwise, related to the information, including but not limited to its condition, quality, performance, merchantability or fitness for purpose. Microchip disclaims all liability arising from this information and its use. Use of Microchip's products as critical components in life support systems is not authorized except with express written approval by Microchip. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights.

The graphics in this document are for illustration only. Microchip reserves the right to modify the contents of its development systems.

Trademarks

The Microchip name and logo, the Microchip logo, Accuron, dsPIC, KEELOQ, KEELOQ logo, microID, MPLAB, PIC, PICmicro, PICSTART, PRO MATE, PowerSmart, rfPIC, and SmartShunt are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

AmpLab, FilterLab, Linear Active Thermistor, Migratable Memory, MXDEV, MXLAB, PS logo, SEEVAL, SmartSensor and The Embedded Control Solutions Company are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Analog-for-the-Digital Age, Application Maestro, CodeGuard, dsPICDEM, dsPICDEM.net, dsPICworks, ECAN, ECONOMONITOR, FanSense, FlexROM, fuzzyLAB, In-Circuit Serial Programming, ICSP, ICEPIC, Mindi, MiWi, MPASM, MPLAB Certified logo, MPLIB, MPLINK, PICkit, PICDEM, PICDEM.net, PICLAB, PICtail, PowerCal, PowerInfo, PowerMate, PowerTool, REAL ICE, rfLAB, rfPICDEM, Select Mode, Smart Serial, SmartTel, Total Endurance, UNI/O, WiperLock and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

All other trademarks mentioned herein are property of their respective companies.

Printed on recycled paper.

Worldwide Sales and Service

AMERICAS

Corporate Office
Tel: 480-792-7200
Technical Support:
http://support.microchip.com

Atlanta
Tel: 678-957-9614

Boston
Tel: 774-760-0087

Chicago
Tel: 630-285-0071

Dallas
Tel: 972-818-7423

Detroit
Tel: 248-538-2250

Kokomo
Tel: 765-864-8360

Los Angeles
Tel: 949-462-9523

Santa Clara
Tel: 408-961-6444

Toronto
Tel: 905-673-0699

ASIA/PACIFIC

Australia
Tel: 61-2-9868-6733

China-Beijing
Tel: 86-10-8528-2100

China-Chengdu
Tel: 86-28-8665-5511

China-Fuzhou
Tel: 86-591-8750-3506

China-Hong Kong SAR
Tel: 852-2401-1200

China-Qingdao
Tel: 86-532-8502-7355

China-Shanghai
Tel: 86-21-5407-5533

China-Shenyang
Tel: 86-24-2334-2829

China-Shenzhen
Tel: 86-755-8203-2660

China-Shunde
Tel: 86-757-2839-5507

China-Wuhan
Tel: 86-27-5980-5300

China-Xian
Tel: 86-29-8833-7250

India-Bangalore
Tel: 91-80-4182-8400

India-New Delhi
Tel: 91-11-4160-8631

India-Pune
Tel: 91-20-2566-1512

Japan-Yokohama
Tel: 81-45-471- 6166

Korea-Gumi
Tel: 82-54-473-4301

Korea-Seoul
Tel: 82-2-554-7200

Malaysia-Penang
Tel: 60-4-646-8870

Philippines-Manila
Tel: 63-2-634-9065

Singapore
Tel: 65-6334-8870

Taiwan-Hsin Chu
Tel: 886-3-572-9526

Taiwan-Kaohsiung
Tel: 886-7-536-4818

Taiwan-Taipei
Tel: 886-2-2500-6610

Thailand-Bangkok
Tel: 66-2-694-1351

EUROPE

Austria-Weis
Tel: 43-7242-2244-39

Denmark-Copenhagen
Tel: 45-4450-2828

France-Paris
Tel: 33-1-69-53-63-20

Germany-Munich
Tel: 49-89-627-144-0

Italy-Milan
Tel: 39-0331-742611

Netherlands-Drunen
Tel: 31-416-690399

Spain-Madrid
Tel: 34-91-708-08-90

UK-Wokingham
Tel: 44-118-921-5869

12/08/06

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona, Gresham, Oregon and Mountain View, California. The Company's quality system processes and procedures are for its PIC® MCUs and dsPIC® DSCs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Microchip Technology Inc.
2355 W. Chandler Blvd. • Chandler, AZ 85224 U.S.A.
Phone: 480-792-7200 • Fax: 480-792-9210
www.microchip.com