Gain and Offset Calibration of the Analog Signal Conditioning (OPAMP) Peripheral

Introduction

Author: Lloyd D. Clark, Ph.D., Microchip Technology Inc.

The Analog Signal Conditioning (OPAMP) peripheral can be configured as a programmable gain amplifier (PGA) using an internal resistor ladder to select a gain value. Because the resistors in the internal resistor ladder have some tolerances and may vary as a function of temperature or other parameters, the actual gain may be slightly different from the selected gain value. In some applications, it may be beneficial to precisely determine the gain and input offset of the PGA. This document explains how to calibrate the gain and offset of the PGA using the microcontroller's (MCU’s) internal digital-to-analog converter (DAC) and analog-to-digital converter (ADC). The DAC is used to control the voltage at the input of the PGA, while the ADC is used to measure the voltage at both the input and output of the PGA. The values measured by the ADC can be used to calculate the gain and input offset of the PGA. No external components are required.

Features

- Gain and Offset Calibration Performed Using the MCU’s Internal DAC and ADC
- Zero External Components Required

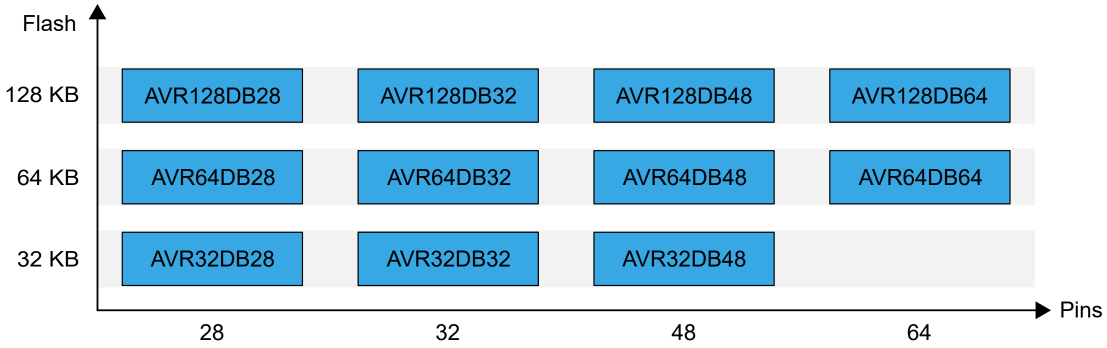
Table of Contents

1. Relevant Devices

This section lists the relevant devices for this document. The following figures show the different family devices, laying out pin count variants and memory sizes:

- Vertical migration upwards is possible without code modification, as these devices are pin-compatible and provide the same or more features
- Horizontal migration to the left reduces the pin count and, therefore, the available features
- Devices with different Flash memory sizes typically also have different SRAM and EEPROM

Figure 1-1. AVR® DB Family Overview

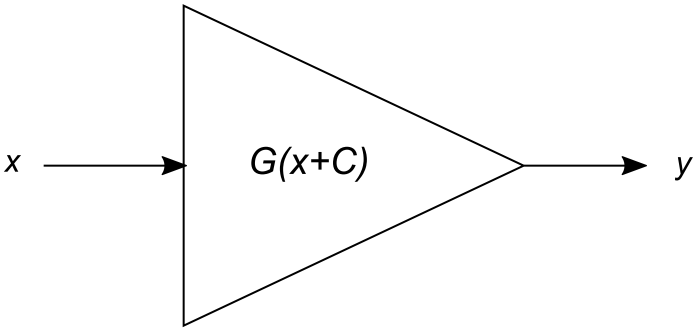


Flash	28	32	48	64
128 KB	AVR128DB28	AVR128DB32	AVR128DB48	AVR128DB64
64 KB	AVR64DB28	AVR64DB32	AVR64DB48	AVR64DB64
32 KB	AVR32DB28	AVR32DB32	AVR32DB48	

2. Calibration Theory

An ideal programmable gain amplifier (PGA) amplifies an input voltage by a precise programmed gain value, *G*. If the PGA is ideal, the voltage measured at the output can be divided by *G* to determine the input voltage. However, many PGA implementations have two imperfections that must be considered in an application. First, there is an input offset voltage that is effectively added to the input signal before amplification. Second, the actual gain may be slightly different from the programmed value due to analog component tolerances, etc.

Figure 2-1. PGA with Gain and Offset Equations



$$y_a = G(x_a+C)$$

$$y_b = G(x_b+C)$$

$$G = (y_b-y_a)/(x_b-x_a)$$

$$C = (y_a/G)-x_a$$

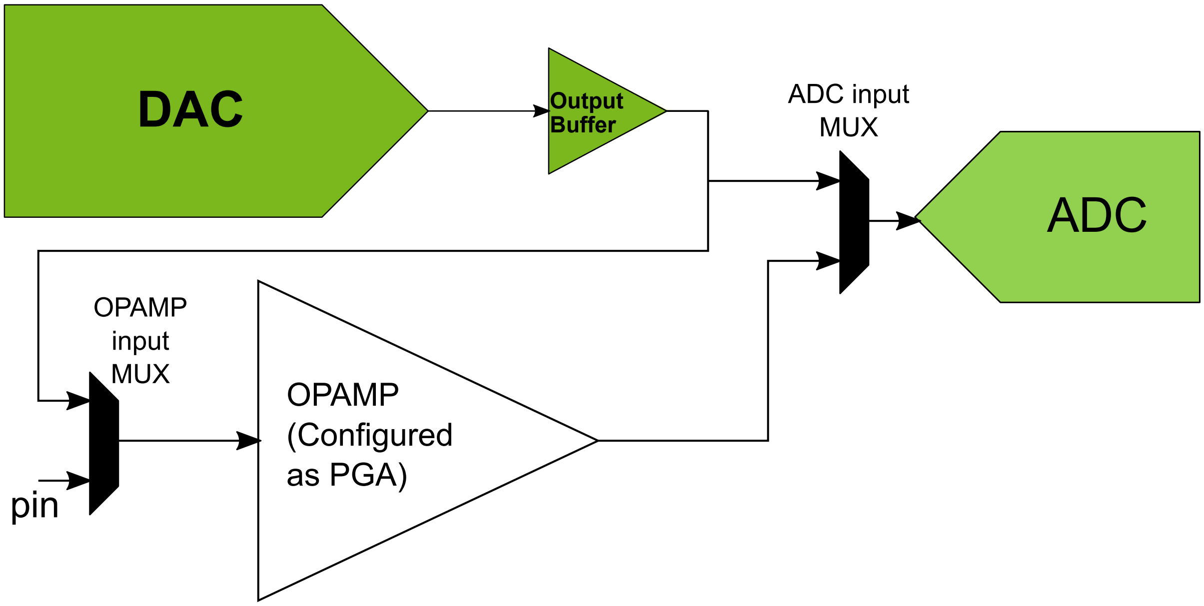
The figure above depicts a high-level representation of a PGA with gain and offset equations included. For an input signal x, an offset value of C is added, and then the sum $(x+C)$ is multiplied by the gain value G to obtain the output signal y. The fundamental idea behind gain and offset calibration is to input two different values, x_a and x_b, so as to obtain two different output values, y_a and y_b. If x_a, x_b, y_a, and y_b are all determined accurately via measurement, the first two equations can be solved for G and C, allowing the gain and input offset to be calculated.

As a realistic example, consider a PGA that has been programmed to have a nominal gain of 16. First, a voltage of 60 mV is input, and the output voltage is measured as 986 mV, so x_a is 60 mV, and y_a is 986 mV. Next, a voltage of 120 mV is input, and the output voltage is measured as 1940 mV, so x_b is 120 mV, and y_b is 1940 mV. Inserting these four values into the equations for G and C leads to a result of 15.9 for the gain G and 2.0 mV for the input offset C.

3. Calibration Implementation on MCU

Because the microcontroller (MCU) contains both a digital-to-analog converter (DAC) and an analog-to-digital converter (ADC), as well as analog multiplexers that allow flexible interconnections, it is possible to perform gain and offset calibration of the programmable gain amplifier (PGA) using no external components.

Figure 3-1. Block Diagram of DAC, OPAMP, and ADC Interconnections Used for Calibration



The figure above shows the DAC, OPAMP, and ADC interconnections that are used for calibration. The buffered DAC output feeds into the ADC input multiplexer and the OPAMP input multiplexer. The OPAMP input multiplexer can select between the buffered DAC output or a pin of the device. The ADC input multiplexer can select between either the DAC output or the OPAMP output.

After the OPAMP, DAC, and ADC have all been initialized and enabled, the following steps are used to implement a gain and offset calibration procedure:

1. Configure the OPAMP input MUX to select the DAC output.
2. Write a first value (x_a) to the DAC.
3. Configure the ADC input MUX to select the DAC output.
4. Allow time for the DAC output and ADC input to settle.
5. Start an ADC conversion.
6. Wait for the ADC conversion to complete.
7. Save the ADC result as x_a.
8. Configure the ADC input MUX to select the OPAMP output.
9. Allow time for the ADC input to settle.
10. Start an ADC conversion.
11. Wait for the ADC conversion to complete.
12. Save the ADC result as y_a.
13. Write a second value (x_b) to the DAC.
14. Configure the ADC input MUX to select the DAC output.
15. Allow time for the DAC output and ADC input to settle.
16. Start an ADC conversion.
17. Wait for the ADC conversion to complete.
18. Save the result of the ADC conversion as x_b.
19. Configure the ADC input MUX to select the OPAMP output.

20. Allow time for the ADC input to settle.
21. Start an ADC conversion.
22. Wait for the ADC conversion to complete.
23. Save the result of the ADC conversion as y_b.
24. Steps 2 through 23 may be repeated many times, and the ADC results averaged to obtain more accurate values for x_a, y_a, x_b, and y_b.
25. Compute the calibrated gain value as $G = (y_b - y_a)/(x_b - x_a)$.
26. Compute the calibrated input offset value as $C = (y_a/G) - x_a$.
27. Calibration is complete, and the OPAMP (configured as PGA) is ready to be used by the application. Configure the OPAMP input MUX to select the device pin.

The calibrated gain and input offset values should be used by the application code for any calculations that involve gain and/or offset.

The code below implements the gain and offset calibration procedure outlined above:

```
#define F_CPU 4000000UL // 4 MHz is default CPU and peripheral frequency

#include <avr/io.h>
#include <util/delay.h>

// The VREF (Voltage Reference) peripheral will be set up to generate a
// reference voltage of 2.5 V
#define VREF_REFSEL_CONTROL VREF_REFSEL_2V500_gc
#define VREF_MV 2500.0 // Floating-point value of reference voltage in mV

// The DAC (Digital-to-Analog Converter) has 10-bit resolution and 2^10 = 1024
// possible input values
// DAC outputs of approximately 60 mV and 120 mV will be used to perform a
// two-point gain and offset measurement/calibration of the op amp gain stage
// Here is the calculation of the two DAC data values needed to generate these
// output voltages:
// (60 mV / 2500 mV) * 1024 = 25
// (120 mV / 2500 mV) * 1024 = 49
#define DAC_DATA_A 25
#define DAC_DATA_B 49

// The AVR-DB ADC (Analog-to-Digital Converter) has 12-bit resolution and
// 2^12 = 4096 possible output values
// Determine the number of mV equivalent to one LSB change in the ADC output:
#define MV_PER_ADC_LSB (VREF_MV/4096)

// Number of measurements to average
#define N_AVERAGE 100

void measure(uint16_t dac_data, uint8_t muxpos_dacout, uint8_t muxpos_opout,
                int16_t *dac_meas, int16_t *opout_meas){

    //Given a DAC setting, this function measures the op amp input and output values

    DAC0.DATA = dac_data << DAC_DATA0_bp; // Write a new value to the DAC
    ADC0.MUXPOS = muxpos_dacout; // Configure the ADC input mux to select the DAC output
    _delay_ms(1); // Allow time for the DAC output and ADC input to settle

    ADC0.COMMAND = ADC_STCONV_bm; // Start an ADC conversion
    while(ADC0.COMMAND & ADC_STCONV_bm); // Wait for the ADC conversion to complete
    *dac_meas = ADC0.RES; // Save the ADC result as a measurement of the DAC output value

    ADC0.MUXPOS = muxpos_opout; // Configure the ADC input mux to select the OPAMP output
    _delay_ms(1); // Allow time for the ADC input to settle

    ADC0.COMMAND = ADC_STCONV_bm; // Start an ADC conversion
    while(ADC0.COMMAND & ADC_STCONV_bm); // Wait for the ADC conversion to complete
    *opout_meas = ADC0.RES; // Save ADC result as a measurement of the OPAMP output

    return;
}

volatile float meas_gain = 0; // Measured gain
volatile float meas_offset = 0; // Measured input offset in ADC units
volatile float meas_offset_mv = 0; // Measured input offset in mV
```

```
int main(void)
{
    uint8_t n;
    int16_t xa, ya, xb, yb;
    int32_t accum_xa, accum_ya, accum_xb, accum_yb;
    float avg_xa, avg_ya, avg_xb, avg_yb;

    // Disable digital inputs on OP0 input and output pins
    PORTD.PIN1CTRL = PORT_ISC_INPUT_DISABLE_gc;
    PORTD.PIN2CTRL = PORT_ISC_INPUT_DISABLE_gc;
    PORTD.PIN3CTRL = PORT_ISC_INPUT_DISABLE_gc;

    // Disable digital input on DAC output pin
    PORTD.PIN6CTRL = PORT_ISC_INPUT_DISABLE_gc;

    // Set up the timebase of the OPAMP peripheral
    OPAMP.TIMEBASE = 3; // Number of CLK_PER cycles that equal one us, minus one (4-1=3)

    //Configure OP0 as non-inverting gain of 16 with DAC as input
    OPAMP.OP0INMUX = OPAMP_OP0INMUX_MUXPOS_DAC_gc | OPAMP_OP0INMUX_MUXNEG_WIP_gc;
    OPAMP.OP0RESMUX = OPAMP_OP0RESMUX_MUXBOT_GND_gc | OPAMP_OP0RESMUX_MUXWIP_WIP7_gc |
OPAMP_OP0RESMUX_MUXTOP_OUT_gc;
    // Configure OP0 Control A
    OPAMP.OP0CTRLA = OPAMP_RUNSTBY_bm |  OPAMP_OP0CTRLA_OUTMODE_NORMAL_gc | OPAMP_ALWAYSON_bm;

    // Enable the OPAMP peripheral
    OPAMP.CTRLA = OPAMP_ENABLE_bm;

    // Set up VREF peripheral to generate the same reference for both ADC and DAC
    VREF.ADC0REF = VREF_ALWAYSON_bm | VREF_REFSEL_CONTROL;
    VREF.DAC0REF = VREF_ALWAYSON_bm | VREF_REFSEL_CONTROL;

    // Set up DAC peripheral by enabling it and its output pin
    DAC0.CTRLA = DAC_OUTEN_bm | DAC_ENABLE_bm;

    // Set up ADC peripheral
    ADC0.CTRLC = ADC_PRESC_DIV4_gc; // Set up ADC prescaler to DIV4
                                    // so CLK_ADC = 4 MHz / 4 = 1 MHz
    ADC0.CTRLA = ADC_ENABLE_bm; // Enable ADC in single-ended mode

    accum_xa = 0; accum_ya = 0; accum_xb = 0; accum_yb = 0; //Reset accumulators
    for (n = 0; n < N_AVERAGE; n++){
        measure(DAC_DATA_A, ADC_MUXPOS_AIN6_gc, ADC_MUXPOS_AIN2_gc, &xa, &ya);
        //                  DACOUT              OP0OUT
        measure(DAC_DATA_B, ADC_MUXPOS_AIN6_gc, ADC_MUXPOS_AIN2_gc, &xb, &yb);
        //                  DACOUT              OP0OUT
        accum_xa += xa; accum_ya += ya; accum_xb += xb; accum_yb += yb; //Add to accumulators
    }
    avg_xa = ((float) accum_xa)/((float) N_AVERAGE);
    avg_ya = ((float) accum_ya)/((float) N_AVERAGE);
    avg_xb = ((float) accum_xb)/((float) N_AVERAGE);
    avg_yb = ((float) accum_yb)/((float) N_AVERAGE);

    meas_gain = (avg_yb - avg_ya)/(avg_xb - avg_xa); // Measured gain
    meas_offset = (avg_ya/meas_gain) - avg_xa; // Measured input offset in ADC units
    meas_offset_mv = meas_offset * MV_PER_ADC_LSB; // Measured input offset in mV

    // Calibration is complete and the OPAMP is ready to be used by the application,
    // so configure the OPAMP input mux to select the input pin.
    OPAMP.OP0INMUX = OPAMP_OP0INMUX_MUXPOS_INP_gc | OPAMP_OP0INMUX_MUXNEG_WIP_gc;

    while(1){
        // Place application here
    }

}
```

View Code Example on GitHub
Click to browse repository

4. Revision History

Doc. Rev.	Date	Comments
A	09/2020	Initial document release

The Microchip Website

Microchip provides online support via our website at www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specifications contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is secure when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods being used in attempts to breach the code protection features of the Microchip devices. We believe that these methods require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Attempts to breach these code protection features, most likely, cannot be accomplished without violating Microchip's intellectual property rights.
- Microchip is willing to work with any customer who is concerned about the integrity of its code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of its code. Code protection does not mean that we are guaranteeing the product is "unbreakable." Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication is provided for the sole purpose of designing with and using Microchip products. Information 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.

THIS INFORMATION IS PROVIDED BY MICROCHIP “AS IS”. 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 ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE OR WARRANTIES RELATED TO ITS CONDITION, QUALITY, OR PERFORMANCE.

IN NO EVENT WILL MICROCHIP BE LIABLE FOR ANY INDIRECT, SPECIAL, PUNITIVE, INCIDENTAL OR CONSEQUENTIAL LOSS, DAMAGE, COST OR EXPENSE OF ANY KIND WHATSOEVER RELATED TO THE INFORMATION OR ITS USE, HOWEVER CAUSED, EVEN IF MICROCHIP HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEEABLE. TO THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL CLAIMS IN ANY WAY RELATED TO THE INFORMATION OR ITS USE WILL NOT EXCEED THE AMOUNT OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THE INFORMATION. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, Adaptec, AnyRate, AVR, AVR logo, AVR Freaks, BesTime, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, HELDO, IGLOO, JukeBlox, KeeLoq, Kleer, LANCheck, LinkMD, maXStylus, maXTouch, MediaLB, megaAVR, Microsemi, Microsemi logo, MOST, MOST logo, MPLAB, OptoLyzer, PackeTime, PIC, picoPower, PICSTART, PIC32 logo, PolarFire, Prochip Designer, QTouch, SAM-BA, SenGenuity, SpyNIC, SST, SST Logo, SuperFlash, Symmetricom, SyncServer, Tachyon, TempTrackr, TimeSource, tinyAVR, UNI/O, Vectron, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

APT, ClockWorks, The Embedded Control Solutions Company, EtherSynch, FlashTec, Hyper Speed Control, HyperLight Load, IntelliMOS, Libero, motorBench, mTouch, Powermite 3, Precision Edge, ProASIC, ProASIC Plus, ProASIC Plus logo, Quiet-Wire, SmartFusion, SyncWorld, Temux, TimeCesium, TimeHub, TimePictra, TimeProvider, Vite, WinPath, and ZL are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BlueSky, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KleerNet, KleerNet logo, memBrain, Mindi, MiWi, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICkit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, 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.

The Adaptec logo, Frequency on Demand, Silicon Storage Technology, and Symmcom are registered trademarks of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

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

© 2020, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-6648-2

Quality Management System

For information regarding Microchip’s Quality Management Systems, please visit www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support:
www.microchip.com/support
Web Address:
www.microchip.com
Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455
Austin, TX
Tel: 512-257-3370
Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088
Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075
Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924
Detroit
Novi, MI
Tel: 248-848-4000
Houston, TX
Tel: 281-894-5983
Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380
Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800
Raleigh, NC
Tel: 919-844-7510
New York, NY
Tel: 631-435-6000
San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270
Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

Australia - Sydney
Tel: 61-2-9868-6733
China - Beijing
Tel: 86-10-8569-7000
China - Chengdu
Tel: 86-28-8665-5511
China - Chongqing
Tel: 86-23-8980-9588
China - Dongguan
Tel: 86-769-8702-9880
China - Guangzhou
Tel: 86-20-8755-8029
China - Hangzhou
Tel: 86-571-8792-8115
China - Hong Kong SAR
Tel: 852-2943-5100
China - Nanjing
Tel: 86-25-8473-2460
China - Qingdao
Tel: 86-532-8502-7355
China - Shanghai
Tel: 86-21-3326-8000
China - Shenyang
Tel: 86-24-2334-2829
China - Shenzhen
Tel: 86-755-8864-2200
China - Suzhou
Tel: 86-186-6233-1526
China - Wuhan
Tel: 86-27-5980-5300
China - Xian
Tel: 86-29-8833-7252
China - Xiamen
Tel: 86-592-2388138
China - Zhuhai
Tel: 86-756-3210040

ASIA/PACIFIC

India - Bangalore
Tel: 91-80-3090-4444
India - New Delhi
Tel: 91-11-4160-8631
India - Pune
Tel: 91-20-4121-0141
Japan - Osaka
Tel: 81-6-6152-7160
Japan - Tokyo
Tel: 81-3-6880- 3770
Korea - Daegu
Tel: 82-53-744-4301
Korea - Seoul
Tel: 82-2-554-7200
Malaysia - Kuala Lumpur
Tel: 60-3-7651-7906
Malaysia - Penang
Tel: 60-4-227-8870
Philippines - Manila
Tel: 63-2-634-9065
Singapore
Tel: 65-6334-8870
Taiwan - Hsin Chu
Tel: 886-3-577-8366
Taiwan - Kaohsiung
Tel: 886-7-213-7830
Taiwan - Taipei
Tel: 886-2-2508-8600
Thailand - Bangkok
Tel: 66-2-694-1351
Vietnam - Ho Chi Minh
Tel: 84-28-5448-2100

EUROPE

Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393
Denmark - Copenhagen
Tel: 45-4485-5910
Fax: 45-4485-2829
Finland - Espoo
Tel: 358-9-4520-820
France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79
Germany - Garching
Tel: 49-8931-9700
Germany - Haan
Tel: 49-2129-3766400
Germany - Heilbronn
Tel: 49-7131-72400
Germany - Karlsruhe
Tel: 49-721-625370
Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44
Germany - Rosenheim
Tel: 49-8031-354-560
Israel - Ra’anana
Tel: 972-9-744-7705
Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781
Italy - Padova
Tel: 39-049-7625286
Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340
Norway - Trondheim
Tel: 47-72884388
Poland - Warsaw
Tel: 48-22-3325737
Romania - Bucharest
Tel: 40-21-407-87-50
Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91
Sweden - Gothenberg
Tel: 46-31-704-60-40
Sweden - Stockholm
Tel: 46-8-5090-4654
UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820