# ADT7473

# Remote Thermal Monitor and Fan Control

The ADT7473/ADT7473−1 controller is a thermal monitor and multiple PWM fan controller for noise sensitive or power sensitive applications requiring active system cooling. The ADT7473/ADT7473−1 can drive a fan using either a low or high frequency drive signal, monitor the temperature of up to two remote sensor diodes plus its own internal temperature, and measure and control the speed of up to four fans so they operate at the lowest possible speed for minimum acoustic noise.

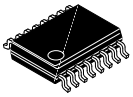
The automatic fan speed control loop optimizes fan speed for a given temperature. A unique dynamic $T_{MIN}$ control mode enables the system thermals/acoustics to be intelligently managed. The effectiveness of the system's thermal solution can be monitored using the $\overline{\text{THERM}}$ input. The ADT7473/ADT7473−1 also provide critical thermal protection to the system using the bidirectional $\overline{\text{THERM}}$ pin as an output to prevent system or component overheating.

### Features

- Controls and Monitors Up to 4 Fans
- High and Low Frequency Fan Drive Signal
- 1 On-Chip and 2 Remote Temperature Sensors
- Series Resistance Cancellation on the Remote Channel
- Extended Temperature Measurement Range, Up to 191°C
- Dynamic $T_{MIN}$ Control Mode Intelligently Optimizes System Acoustics
- Automatic Fan Speed Control Mode Controls System Cooling Based on Measured Temperature
- Enhanced Acoustic Mode Dramatically Reduces User Perception of Changing Fan Speeds
- Thermal Protection Feature via $\overline{\text{THERM}}$ Output
- Monitors Performance Impact of Intel® Pentium® 4 Processor
- Thermal Control Circuit via $\overline{\text{THERM}}$ Input
- 3-wire and 4-wire Fan Speed Measurement
- Limit Comparison of All Monitored Values
- Meets SMBus 2.0 Electrical Specifications (Fully SMBus 1.1 Compliant)
- These Devices are Pb-Free, Halogen Free/BFR Free and are RoHS Compliant

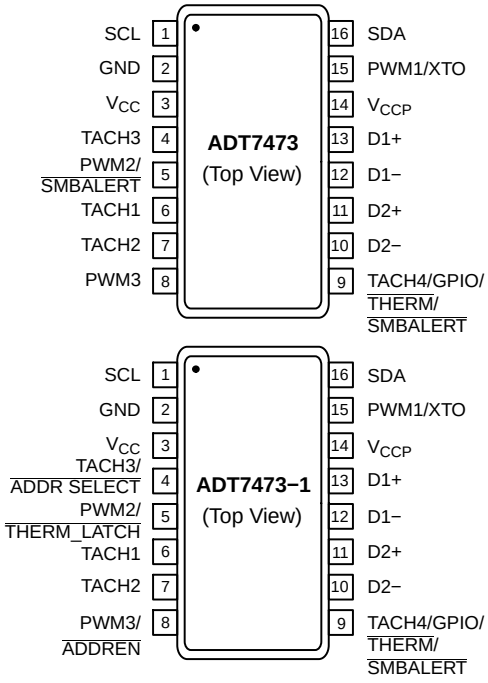
ON Semiconductor®

www.onsemi.com

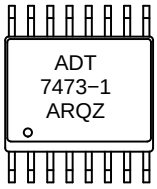


**QSOP−16**
**CASE 492**

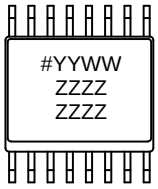
### PIN ASSIGNMENTS




### TOP MARKING




### BOTTOM MARKING




ADT7473−1ARQZ = Specific Device Code
\# = Pb-Free Package
YYWW = Date Code
ZZZZ = Assembly Lot Code

### ORDERING INFORMATION

See detailed ordering and shipping information in the package dimensions section on page 72 of this data sheet.

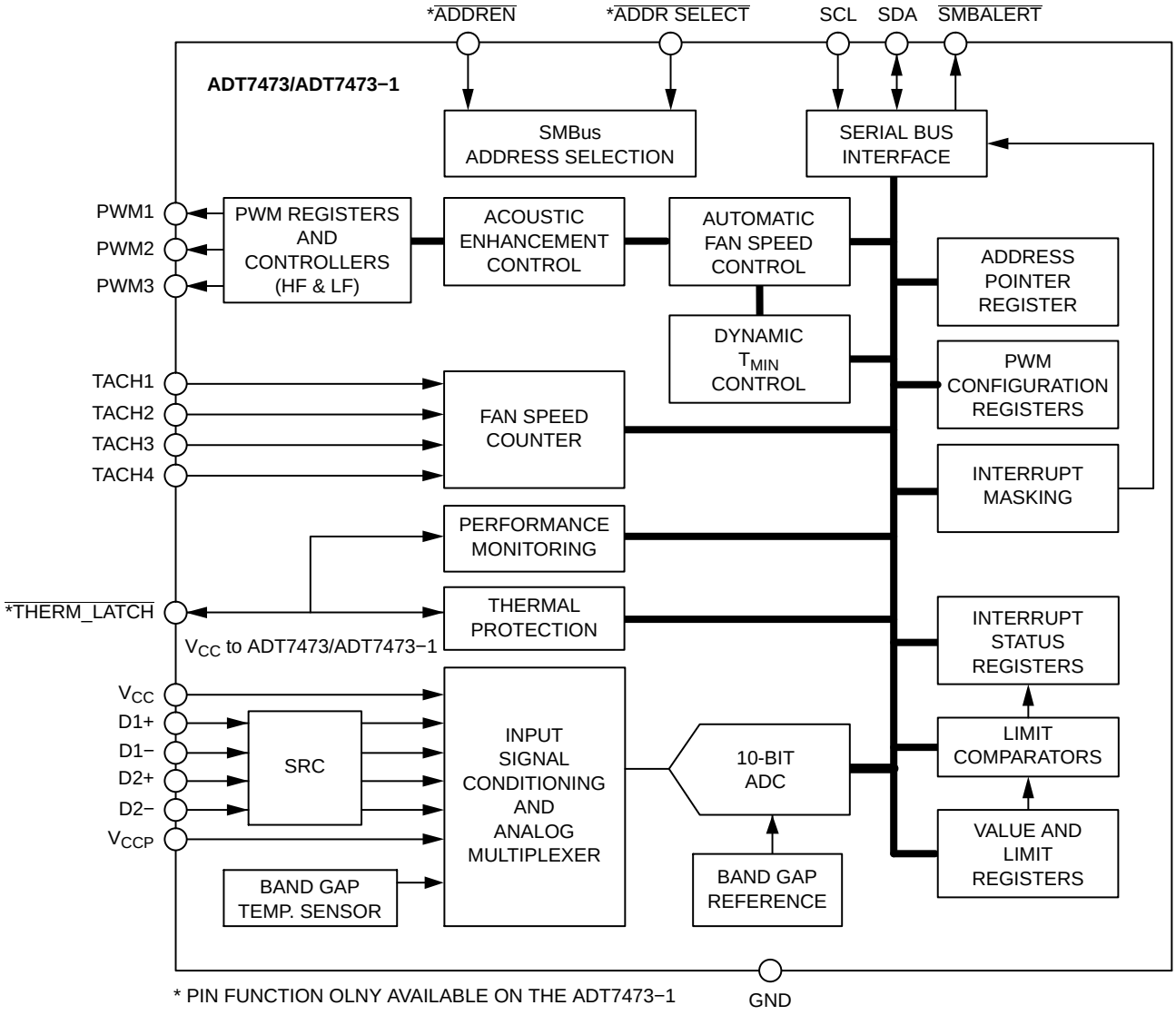



**Figure 1. Functional Block Diagram**

**Table 1. ABSOLUTE MAXIMUM RATINGS**

| Parameter | Rating | Unit |
|---|---|---|
| Positive Supply Voltage ($V_{CC}$) | 3.6 | V |
| Voltage on any Input or Output Pin | −0.3 to +3.6 | V |
| Input Current at any Pin | ±5.0 | mA |
| Package Input Current | ±20 | mA |
| Maximum Junction Temperature ($T_{J\ MAX}$) | 150 | °C |
| Storage Temperature Range | −65 to +150 | °C |
| Lead Temperature, Soldering<br>IR Reflow Peak Temperature<br>Lead Temperature (Soldering, 10 sec) | <br>260<br>300 | °C |
| ESD Rating | 1,500 | V |

Stresses exceeding those listed in the Maximum Ratings table may damage the device. If any of these limits are exceeded, device functionality should not be assumed, damage may occur and reliability may be affected.

NOTE: This device is ESD sensitive. Use standard ESD precautions when handling.

**Table 2. THERMAL CHARACTERISTICS**

| Package Type | $\theta_{JA}$ | $\theta_{JC}$ | Unit |
|---|---|---|---|
| 16-lead QSOP | 150 | 39 | °C/W |

NOTE: $\theta_{JA}$ is specified for the worst-case conditions, that is, a device soldered in a circuit board for surface-mount packages.

**Table 3. PIN ASSIGNMENT**

| Pin No. | Mnemonic | Description |
|---|---|---|
| 1 | SCL | Digital Input (Open Drain). SMBus serial clock input. Requires SMBus pullup. |
| 2 | GND | Ground Pin. |
| 3 | $V_{CC}$ | Power Supply. Powered by 3.3 V. |
| 4 | TACH3<br>$\overline{\text{ADDR SELECT}}$ | Digital Input (Open Drain). Fan tachometer input to measure speed of Fan 3.<br>If in address select mode, the logic state of this pin defines the SMBus device address. |
| 5 | PWM2<br><br>$\overline{\text{SMBALERT}}$<br><br>$\overline{\text{THERM\_LATCH}}$ | Digital Output (Open Drain). ADT7473 default pin function is PWM2. Requires 10 kΩ typical pullup. Pulse-width modulated output to control Fan 2 speed. Can be configured as a high or low frequency drive.<br>On the ADT7473, this pin can be reconfigured as an $\overline{\text{SMBALERT}}$ interrupt output to signal out-of-limit conditions.<br>ADT7473−1 default pin function. $\overline{\text{THERM\_LATCH}}$ is a thermal event alert signal when an overtemperature condition occurs. |
| 6 | TACH1 | Digital Input (Open Drain). Fan tachometer input to measure speed of Fan 1. |
| 7 | TACH2 | Digital Input (Open Drain). Fan tachometer input to measure speed of Fan 2. |
| 8 | PWM3<br><br>$\overline{\text{ADDREN}}$ | Digital I/O (Open Drain). Pulse-width modulated output to control the speed of Fan 3 and Fan 4. Requires 10 kΩ typical pullup. Can be configured as a high or low frequency drive.<br>If pulled low on powerup, the ADT7473−1 enters address select mode, and the state of Pin 4 ($\overline{\text{ADDR SELECT}}$) determines the ADT7473−1 slave address. |
| 9 | TACH4<br>GPIO<br>$\overline{\text{THERM}}$<br><br><br>$\overline{\text{SMBALERT}}$ | Digital Input (Open Drain). Fan tachometer input to measure speed of Fan 4.<br>General Purpose Open Drain Digital I/O.<br>Bidirectional $\overline{\text{THERM}}$ pin. Can be used to time and monitor assertions on the $\overline{\text{THERM}}$ input as well as to assert when an ADT7473 $\overline{\text{THERM}}$ overtemperature limit is exceeded. For example, the pin can be connected to the $\overline{\text{PROCHOT}}$ output of an Intel® Pentium® 4 processor or to the output of a trip point temperature sensor. Can be used as an output to signal overtemperature conditions.<br>Digital Output (Open Drain). This pin can be reconfigured as an $\overline{\text{SMBALERT}}$ interrupt output to signal out-of-limit conditions. |
| 10 | D2− | Cathode Connection to Second Thermal Diode. |
| 11 | D2+ | Anode Connection to Second Thermal Diode. |
| 12 | D1− | Cathode Connection to First Thermal Diode. |
| 13 | D1+ | Anode Connection to First Thermal Diode. |
| 14 | $V_{CCP}$ | Analog Input. Monitors processor core voltage (0 V to 3.0 V). |
| 15 | PWM1<br><br>XTO | Digital Output (Open Drain). Pulse-width modulated output to control Fan 1 speed. Requires 10 kΩ typical pullup.<br>Also functions as the output from the XNOR tree in XNOR test mode. |
| 16 | SDA | Digital I/O (Open Drain). SMBus bidirectional serial data. Requires 10 kΩ typical pullup. |

**Table 4. ELECTRICAL CHARACTERISTICS** ($T_A = T_{MIN}$ to $T_{MAX}$, $V_{CC} = V_{MIN}$ to $V_{MAX}$, unless otherwise noted.) (Note 1)

| Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| **Power Supply** | | | | | |
| Supply Voltage | | 3.0 | 3.3 | 3.6 | V |
| Supply Current, $I_{CC}$ | Interface Inactive, ADC Active | − | 1.5 | 3.0 | mA |
| **Temperature-to-Digital Converter** | | | | | |
| Local Sensor Accuracy<br><br>Resolution | 0°C ≤ $T_A$ ≤ 85°C<br>−40°C ≤ $T_A$ ≤ +125°C | −<br>−<br>− | ±0.5<br>−<br>0.25 | ±1.5<br>±2.5<br>− | °C |
| Remote Diode Sensor Accuracy<br><br>Resolution | 0°C ≤ $T_A$ ≤ 85°C<br>−40°C ≤ $T_A$ ≤ +125°C | −<br>−<br>− | ±0.5<br>−<br>0.25 | ±1.5<br>±2.5<br>− | °C |
| Remote Sensor Source Current | First Current<br>Second Current<br>Third Current | −<br>−<br>− | 6<br>36<br>96 | −<br>−<br>− | μA |
| **Analog-to-Digital Converter (Including MUX and Attentuators)** | | | | | |
| Total Unadjusted Error (TUE) | | − | − | ±1.5 | % |
| Differential Nonlinearity (DNL) | 8 Bits | − | − | ±1.0 | LSB |
| Power Supply Sensitivity | | − | ±0.1 | − | %/V |
| Conversion Time (Voltage Input) | Averaging Enabled | − | 11 | − | ms |
| Conversion Time (Local Temperature) | Averaging Enabled | − | 12 | − | ms |
| Conversion Time (Remote Temperature) | Averaging Enabled | − | 38 | − | ms |
| Total Monitoring Cycle Time | Averaging Enabled<br>Averaging Disabled | −<br>− | 145<br>19 | −<br>− | ms |
| Input Resistance | For $V_{CCP}$ Channel | 70 | 120 | − | kΩ |
| **Fan RPM-to-Digital Converter** | | | | | |
| Accuracy | 0°C ≤ $T_A$ ≤ 70°C<br>−40°C ≤ $T_A$ ≤ +120°C | −<br>− | −<br>− | ±6.0<br>±10 | % |
| Full-scale Count | | − | − | 65,535 | |
| Nominal Input RPM | Fan Count = 0xBFFF<br>Fan Count = 0x3FFF<br>Fan Count = 0x0438<br>Fan Count = 0x021C | −<br>−<br>−<br>− | 109<br>329<br>5,000<br>10,000 | −<br>−<br>−<br>− | RPM |
| **Open-Drain Digital Outputs, PWM1 to PWM3, XTO** | | | | | |
| Current Sink, $I_{OL}$ | | − | − | 8.0 | mA |
| Output Low Voltage, $V_{OL}$ | $I_{OUT}$ = −8.0 mA | − | − | 0.4 | V |
| High Level Output Current, $I_{OH}$ | $V_{OUT} = V_{CC}$ | − | 0.1 | 20 | μA |
| **Open-Drain Serial Data Bus Output (SDA)** | | | | | |
| Output Low Voltage, $V_{OL}$ | $I_{OUT}$ = −4.0 mA | − | − | 0.4 | V |
| High Level Output Current, $I_{OH}$ | $V_{OUT} = V_{CC}$ | − | 0.1 | 1.0 | μA |
| **Digital Output Logic Levels, ADT7473−1 ($\overline{\text{THERM\_LATCH}}$) ADTL+** | | | | | |
| Output High Voltage, $V_{OH}$ | | 0.75 × $V_{CC}$ | − | − | V |
| Output Low Voltage, $V_{OL}$ | | − | − | 0.4 | V |
| **SMBus Digital Inputs (SCL, SDA)** | | | | | |
| Input High Voltage, $V_{IH}$ | | 2.0 | − | − | V |
| Input Low Voltage, $V_{IL}$ | | − | − | 0.4 | V |
| Hysteresis | | − | 500 | − | mV |

**Table 4. ELECTRICAL CHARACTERISTICS** ($T_A = T_{MIN}$ to $T_{MAX}$, $V_{CC} = V_{MIN}$ to $V_{MAX}$, unless otherwise noted.) (Note 1)

| Parameter | Conditions | Min | Typ | Max | Unit |
|---|---|---|---|---|---|
| **Digital Input Logic Levels (TACH Inputs)** | | | | | |
| Input High Voltage, $V_{IH}$ | Maximum Input Voltage | 2.0<br>– | –<br>– | –<br>3.6 | V |
| Input Low Voltage, $V_{IL}$ | Minimum Input Voltage | –<br>−0.3 | –<br>– | 0.8<br>– | V |
| Hysteresis | | – | 0.5 | – | V p-p |
| **Digital Input Logic Levels ($\overline{\text{THERM}}$) ADTL+** | | | | | |
| Input High Voltage, $V_{IH}$ | | $0.75 \times V_{CC}$ | – | – | V |
| Input Low Voltage, $V_{IL}$ | | – | – | 0.8 | V |
| Input High Voltage, $V_{IH}$ | | – | – | – | |
| Input Low Voltage, $V_{IL}$ | $V_{IN} = V_{CC}$ | – | ±1 | – | μA |
| Input Low Current, $I_{IL}$ | $V_{IN} = 0$ | – | ±1 | – | μA |
| Input Capacitance, $C_{IN}$ | | – | 5.0 | – | pF |
| **Serial Bus Timing** (Note 2) (See Figure 2) | | | | | |
| Clock Frequency, $f_{SCLK}$ | | 10 | – | 400 | kHz |
| Glitch Immunity, $t_{SW}$ | | – | – | 50 | ns |
| Bus Free Time, $t_{BUF}$ | | 4.7 | – | – | μs |
| SCL Low Time, $t_{LOW}$ | | 4.7 | – | – | μs |
| SCL High Time, $t_{HIGH}$ | | 4.0 | – | 50 | μs |
| SCL, SDA Rise Time, $t_r$ | | – | – | 1,000 | ns |
| SCL, SDA Fall Time, $t_f$ | | – | – | 300 | μs |
| Data Setup Time, $t_{SU;\,DAT}$ | | 250 | – | – | ns |
| Detect Clock Low Timeout, $t_{TIMEOUT}$ | Can be Optionally Disabled | 15 | – | 35 | ms |

1. All voltages are measured with respect to GND, unless otherwise noted. Typicals are at $T_A = 25°C$ and represent most likely parametric norm. Logic inputs accept input high voltages up to $V_{MAX}$, even when the device is operating down to $V_{MIN}$. Timing specifications are tested at logic levels of $V_{IL} = 0.8$ V for a falling edge and $V_{IH} = 2.0$ V for a rising edge.
2. Serial management bus (SMBus) timing specifications are guaranteed by design and are not production tested.

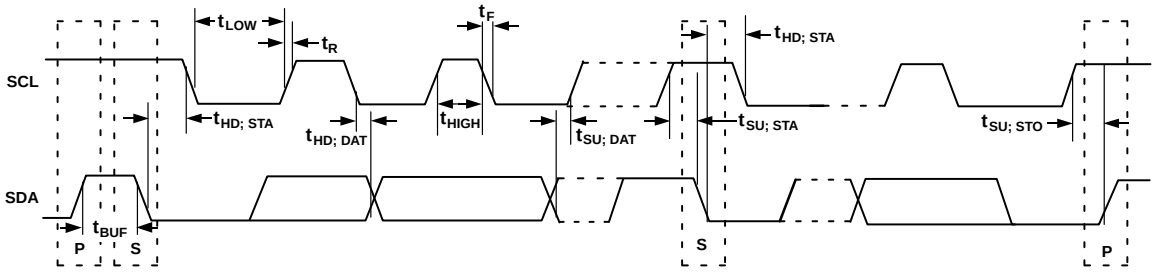



**Figure 2. Serial Bus Timing Diagram**

## TYPICAL PERFORMANCE CHARACTERISTICS

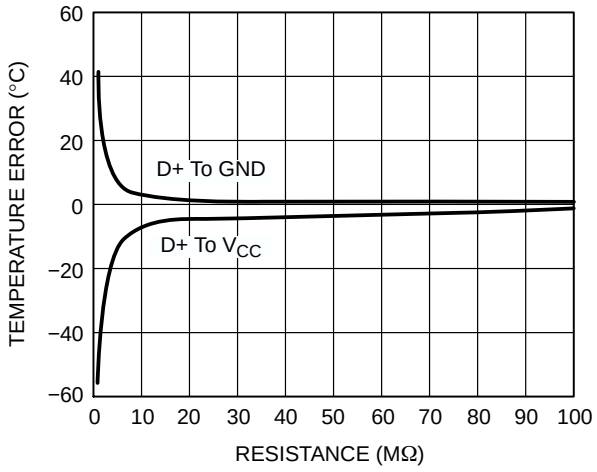



**Figure 3. Remote Temperature Error vs. PCB Resistance**

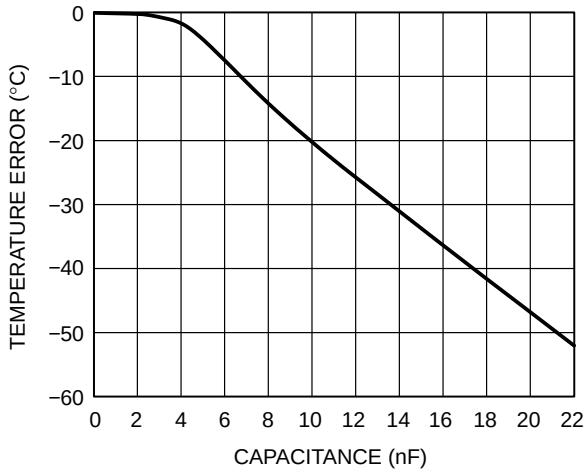



**Figure 4. Temperature Error vs. Capacitance Between D+ and D−**

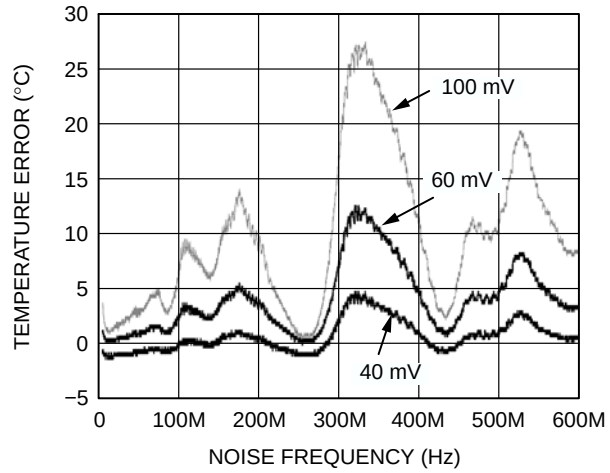



**Figure 5. Remote Temperature Error vs. Common-Mode Noise Frequency**

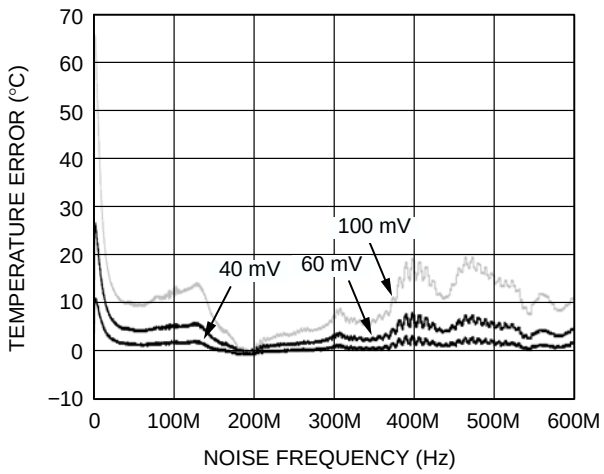



**Figure 6. Remote Temperature Error vs. Common-Mode Noise Frequency**

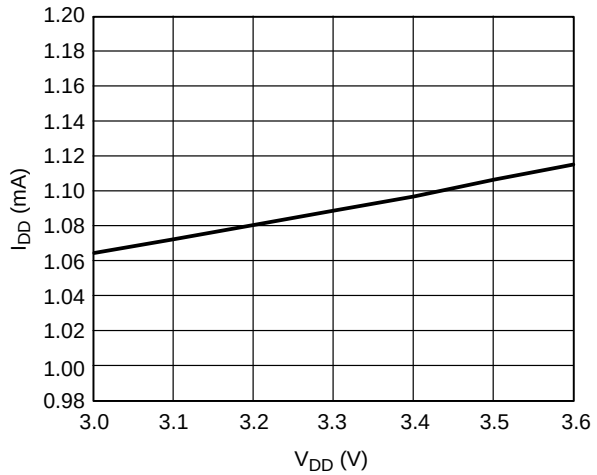



**Figure 7. Normal $I_{DD}$ vs. Power Supply**

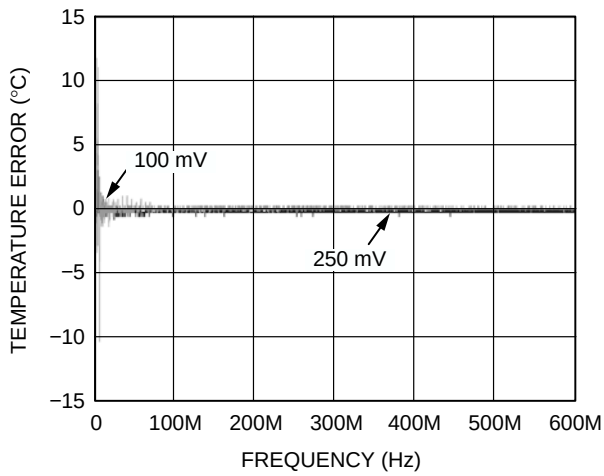



**Figure 8. Internal Temperature Error vs. Frequency**

## TYPICAL PERFORMANCE CHARACTERISTICS (Cont'd)

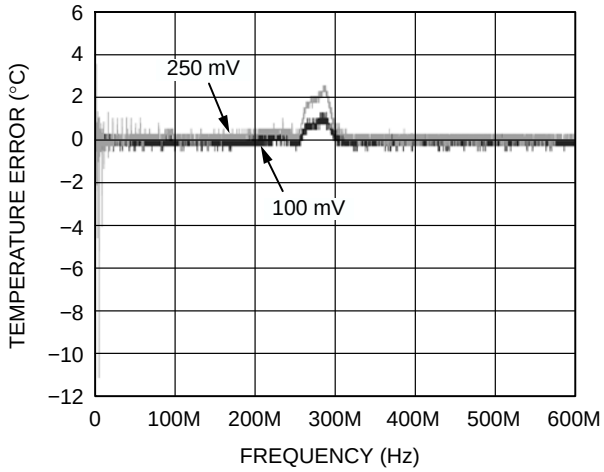



**Figure 9. Remote Temperature Error vs. Power Supply Noise Frequency**

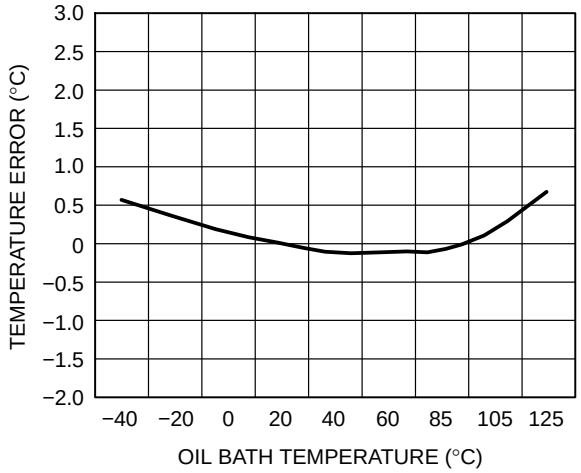



**Figure 10. Internal Temperature Error vs. Temperature**

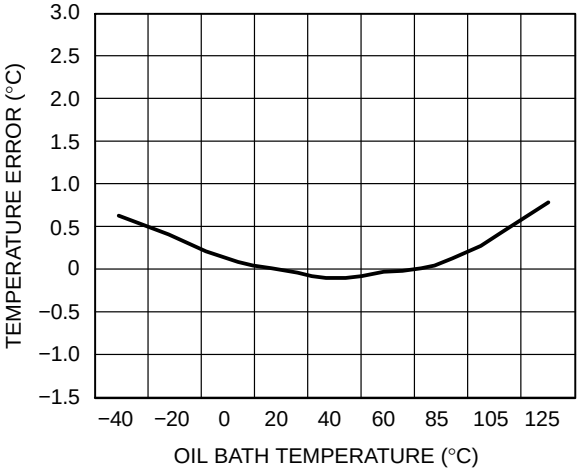



**Figure 11. Remote Temperature Error vs. Temperature**

### Product Description

The ADT7473/ADT7473−1 is a complete thermal monitor and multiple fan controller for any system requiring thermal monitoring and cooling. The device communicates with the system via a serial system management bus. The serial bus controller has a serial data line for reading and writing addresses and data (Pin 16), and an input line for the serial clock (Pin 1). All control and programming functions for the ADT7473/ADT7473−1 are performed over the serial bus. Additionally, a pin can be reconfigured as an $\overline{\text{SMBALERT}}$ output to signal out-of-limit conditions.

Table 5 illustrates the differences between the ADT7473 and the ADT7473−1.

**Table 5. ADT7473/ADT7473−1 DEVICE COMPARISON**

| Feature | ADT7473 | ADT7473−1 |
|---|---|---|
| Pin 5 | Default: PWM2 | Default: $\overline{\text{THERM\_LATCH}}$ |
| SMBus Address | Fixed Address | Address selectable |
| Remote Ch. 2 Therm Limit | = 100°C | = 136°C |
| Register 0x30, 0x31, 0x32 | Default: 0x00 | Default: 0xFF |
| Register 0x3F Revision Reg | Default: 0x68 | Default: 0x69 |
| Register 0x40, Bit 7 | Reserved | (R/W) 1 = Reset Latch (Lockable) |
| Register 0x42, Bit 0 | Reserved | (Read-only) 1 = $\overline{\text{THERM}}$ Limit Latched |
| Registers 0x5C, 0x5D, 0x5E | Default: 0x82 | Default: 0x62 |
| Register 0x7C, Bit 4 | Reserved | $\overline{\text{THERM}}$ Output Hysteresis |
| Register 0x7D, Bit 4 | Reserved | $\overline{\text{THERM\_LATCH}}$ Configuration<br>0 = Remote Channel 2<br>1 = Remote Channel 1 and Remote Channel 2 |

### Comparison Between ADT7467 and ADT7473/ADT7473−1

The following list shows some comparisons between the ADT7467 and the ADT7473/ADT7473−1:

- The ADT7473/ADT7473−1 can be powered via a 3.3 V supply only, and does not support 5.0 V operation, while the ADT7467 does. Violating this specification results in irreversible damage to the ADT7473/ADT7473−1. See the Specifications section for more information.
- A high frequency PWM drive can be independently selected for each PWM channel on the ADT7473/ADT7473−1. This is not available on the ADT7467.
- The range and resolution of the temperature offset register can be changed from a ±64°C range at 0.5°C resolution to a ±128°C range at 1°C resolution. This is not available on the ADT7467.
- $\overline{\text{THERM}}$ overtemperature events can be disabled/enabled individually on each temperature channel. This is not available on the ADT7467.
- Bit 7 of Configuration Register 1 is no longer supported because the ADT7473/ADT7473−1 cannot be powered via a 5.0 V supply.
- Bit 0 of Configuration Register 1 (0x40) remains writable after the lock bit is set. This bit enables monitoring.
- 2-wire fan speed measurement is not supported on the ADT7473/ADT7473−1.

### How to Set the Functionality of Pin 9

Pin 9 on the ADT7473/ADT7473−1 has four possible functions: $\overline{\text{SMBALERT}}$, $\overline{\text{THERM}}$, GPIO, and TACH4. The user chooses the required functionality by setting Bit 0 and Bit 1 of Configuration Register 4 (0x7D).

**Table 6. PIN 9 SETTINGS**

| Bit 0 | Bit 1 | Function |
|---|---|---|
| 0 | 0 | TACH4 |
| 0 | 1 | $\overline{\text{THERM}}$ |
| 1 | 0 | GPIO |
| 1 | 1 | $\overline{\text{SMBALERT}}$ |

### Recommended Implementation

Configuring the ADT7473 as shown in Figure 12 allows the system designer to use the following features:

- Two PWM outputs for fan control of up to three fans. (The front and rear chassis fans are connected in parallel.)
- Three TACH fan speed measurement inputs.
- $V_{CC}$ measured internally through Pin 3.
- CPU temperature measured using Remote 1 temperature channel.
- Ambient temperature measured through Remote 2 temperature channel.
- Bidirectional $\overline{\text{THERM}}$ pin. This feature allows Intel® Pentium® 4 $\overline{\text{PROCHOT}}$ monitoring and can function as an overtemperature $\overline{\text{THERM}}$ output. It can alternatively be programmed as an $\overline{\text{SMBALERT}}$ system interrupt output.

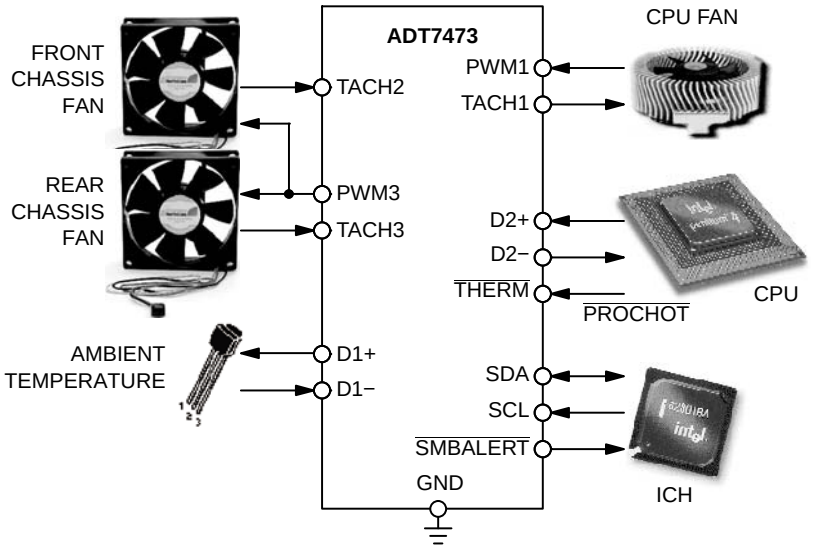



**Figure 12. ADT7473 Configuration**

### Serial Bus Interface

On PCs and servers, control of the ADT7473/ADT7473−1 is carried out using the SMBus. The ADT7473/ADT7473−1 is connected to this bus as a slave device, under the control of a master controller, which is usually (but not necessarily) the ICH.

The ADT7473 has a fixed 7-bit serial bus address of 0101110 or 0x2E. The read/write bit must be added to get the 8-bit address (01011100 or 0x5C). When the ADT7473−1 is powered up with Pin 8 (PWM3/$\overline{\text{ADDREN}}$) high, the ADT7473−1 has a default SMBus address of 0101110 or 0x2E. If more than one ADT7473−1 is used in a system, each ADT7473−1 is placed in $\overline{\text{ADDR SELECT}}$ mode by strapping Pin 8 low on powerup. The logic state of Pin 4 then determines the device's SMBus address. The logic of these pins is sampled on powerup.

The device address is sampled on powerup and latched on the first valid SMBus transaction, more precisely on the low-to-high transition at the beginning of the eighth SCL pulse, when the serial bus address byte matches the selected slave address. The selected slave address is chosen using the $\overline{\text{ADDREN}}$ pin/$\overline{\text{ADDR SELECT}}$ pin. Any attempted change in the address has no effect after this.

**Table 7. HARDWIRING THE ADT7473−1 SMBUS DEVICE ADDRESS**

| Pin 13 State | Pin 14 State | Address |
|---|---|---|
| 0 | Low (10 kΩ to GND) | 0101100 (0x2C) |
| 0 | High (10 kΩ Pullup) | 0101101 (0x2D) |
| 1 | Don't Care | 0101110 (0x2E) |

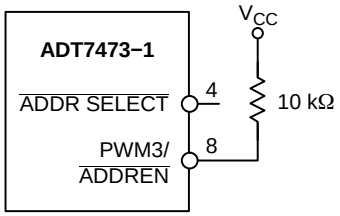



**Figure 13. Default SMBus Address = 0x2E**

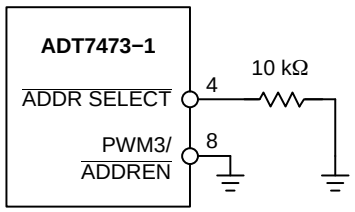



**Figure 14. SMBus Address = 0x2C (Pin 4 = 0)**

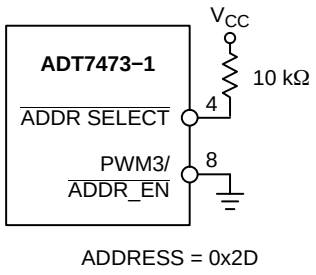
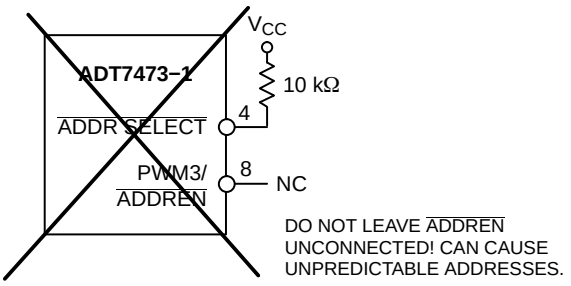



**Figure 15. SMBus Address = 0x2D (Pin 4 = 1)**




CARE SHOULD BE TAKEN TO ENSURE THAT PIN 8 (PWM3/ADDREN) IS EITHER TIED HIGH OR LOW. LEAVING PIN 8 FLOATING COULD CAUSE THE ADT7473−1 TO POWER UP WITH AN UNEXPECTED ADDRESS.

NOTE THAT IF THE ADT7473−1 IS PLACED INTO ADDR SELECT MODE, PINS 8 AND 4 CAN BE USED AS THE ALTERNATE FUNCTIONS (PWM3, TACH4/THERM) UNLESS THE CORRECT CIRCUIT IS MUXED IN AT THE CORRECT TIME OR DESIGNED TO HANDLE THESE DUAL FUNCTIONS.

**Figure 16. Unpredictable SMBus Address if Pin 8 is Unconnected**

The ability to make hardwired changes to the SMBus slave address allows the user to avoid conflicts with other devices sharing the same serial bus, for example, if more than one ADT7473−1 is used in a system.

Data is sent over the serial bus in sequences of nine clock pulses: eight bits of data followed by an acknowledge bit from the slave device. Transitions on the data line must occur during the low period of the clock signal and remain stable during the high period because a low-to-high transition when the clock is high might be interpreted as a stop signal. The number of data bytes that can be transmitted over the serial bus in a single read or write operation is limited only by what the master and slave devices can handle.

When all data bytes have been read or written, stop conditions are established. In write mode, the master pulls the data line high during the tenth clock pulse to assert a stop condition. In read mode, the master device overrides the acknowledge bit by pulling the data line high during the low period before the ninth clock pulse; this is known as No Acknowledge. The master takes the data line low during the low period before the tenth clock pulse, and then high during the tenth clock pulse to assert a stop condition.

Any number of bytes of data can be transferred over the serial bus in one operation, but it is not possible to mix read and write in one operation, because the type of operation is determined at the beginning and cannot subsequently be changed without starting a new operation.

In the ADT7473/ADT7473−1, write operations contain either one or two bytes, and read operations contain one byte. To write data to one of the device data registers or read data from it, the address pointer register must be set so the correct data register is addressed, and then data can be written into that register or read from it. The first byte of a write operation always contains an address that is stored in the address pointer register. If data is written to the device, the write operation contains a second data byte that is written to the register selected by the address pointer register.

This write operation is shown in Figure 17. The device address is sent over the bus, and then R/$\overline{W}$ is set to 0. This is followed by two data bytes. The first data byte is the address of the internal data register to be written to, which is stored in the address pointer register. The second data byte is the data to be written to the internal data register.

When reading data from a register, there are two possibilities:

1. If the ADT7473/ADT7473−1's address pointer register value is unknown or not the desired value, it must first be set to the correct value before data can be read from the desired data register. This is done by performing a write to the ADT7473/ADT7473−1, but only the data byte containing the register address is sent, because no data is written to the register. This is shown in Figure 18.
   A read operation is then performed consisting of the serial bus address, R/$\overline{W}$ bit set to 1, followed by the data byte read from the data register. This is shown in Figure 19.
2. If the address pointer register is known to be already at the desired address, data can be read from the corresponding data register without first writing to the address pointer register, as shown in Figure 19.

It is possible to read a data byte from a data register without first writing to the address pointer register, if the address pointer register is already at the correct value. However, it is not possible to write data to a register without writing to the address pointer register, because the first data byte of a write is always written to the address pointer register.

In addition to supporting the send byte and receive byte protocols, the ADT7473/ADT7473−1 also supports the read byte protocol. (See System Management Bus (SMBus) Specifications Version 2 for more information; this document is available from Intel.)

If several read or write operations must be performed in succession, the master can send a repeat start condition instead of a stop condition to begin a new operation.

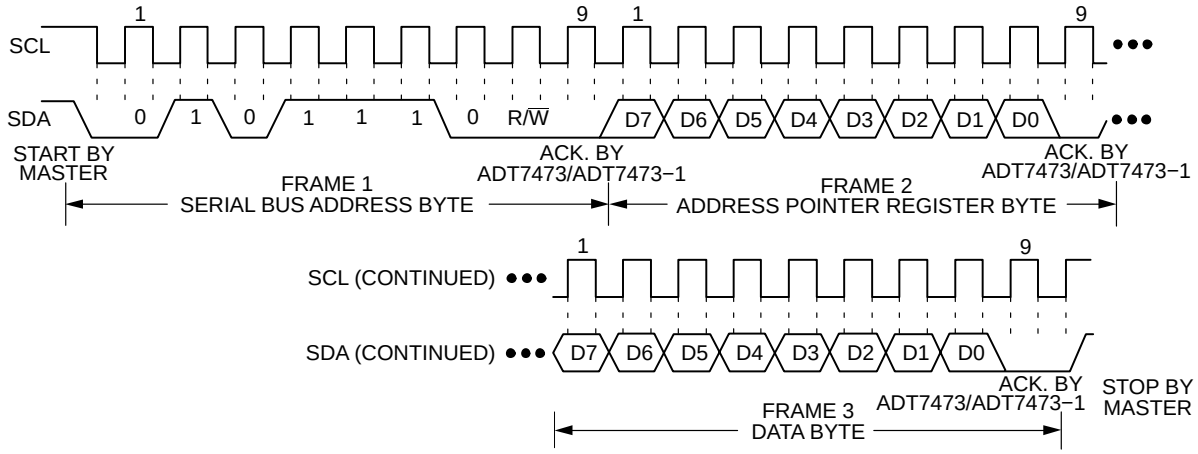



**Figure 17. Writing a Register Address to the Address Pointer Register, then Writing Data to the Selected Register**

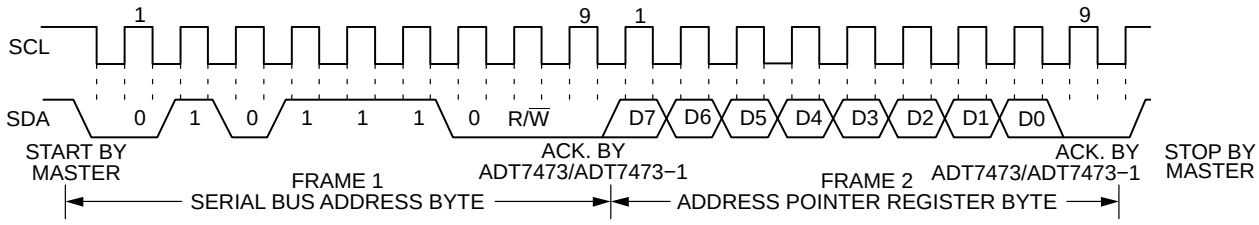



**Figure 18. Writing to the Address Pointer Register Only**

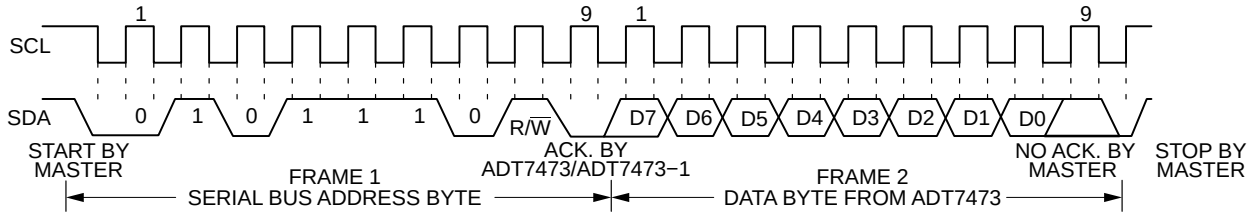



**Figure 19. Reading Data from a Previously Selected Register**

### Write Operations

The SMBus specification defines several protocols for various read and write operations. The ADT7473/ADT7473−1 uses the following SMBus write protocols. The following abbreviations are used in the diagrams:

S − Start
P − Stop
R − Read
W − Write
A − Acknowledge
$\overline{A}$ − No Acknowledge

### Send Byte

In this operation, the master device sends a single command byte to a slave device, as follows:

1. The master device asserts a start condition on SDA.
2. The master sends the 7-bit slave address followed by the write bit (active low).
3. The addressed slave device asserts ACK on SDA.
4. The master sends a command code.
5. The slave asserts ACK on SDA.
6. The master asserts a stop condition on SDA and the transaction ends.

For the ADT7473/ADT7473−1, the send byte protocol is used to write a register address to RAM for a subsequent single-byte read from the same address. This operation is illustrated in Figure 20.

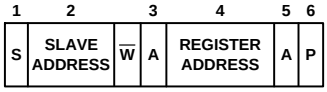



**Figure 20. Setting a Register Address for Subsequent Read**

If the master is required to read data from the register immediately after setting up the address, it can assert a repeat start condition immediately after the final ACK and carry out a single-byte read without asserting an intermediate stop condition.

### Write Byte

In this operation, the master device sends a command byte and one data byte to the slave device, as follows:

1. The master device asserts a start condition on SDA.
2. The master sends the 7-bit slave address followed by the write bit (active low).
3. The addressed slave device asserts ACK on SDA.
4. The master sends a command code.
5. The slave asserts ACK on SDA.
6. The master sends a data byte.
7. The slave asserts ACK on SDA.
8. The master asserts a stop condition on SDA, and the transaction ends.

The single byte write operation is illustrated in Figure 21.

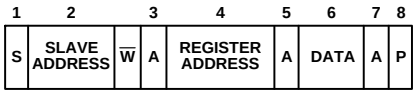



**Figure 21. Single-byte Write to a Register**

### Read Operations

The ADT7473/ADT7473−1 uses the following SMBus read protocols.

### Receive Byte

This operation is useful when repeatedly reading a single register. The register address must have been previously set up. In this operation, the master device receives a single byte from a slave device, as follows:

1. The master device asserts a start condition on SDA.
2. The master sends the 7-bit slave address followed by the read bit (high).
3. The addressed slave device asserts ACK on SDA.
4. The master receives a data byte.
5. The master asserts NO ACK on SDA.
6. The master asserts a stop condition on SDA, and the transaction ends.

In the ADT7473/ADT7473−1, the receive byte protocol is used to read a single byte of data from a register whose address has previously been set by a send byte or write byte operation. This operation is illustrated in Figure 22.

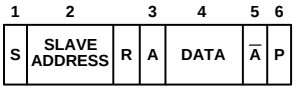



**Figure 22. Single-byte Read from a Register**

### Alert Response Address

Alert response address (ARA) is a feature of SMBus devices that allows an interrupting device to identify itself to the host when multiple devices exist on the same bus.

The $\overline{\text{SMBALERT}}$ output can be used as either an interrupt output or an $\overline{\text{SMBALERT}}$. One or more outputs can be connected to a common $\overline{\text{SMBALERT}}$ line connected to the master. If a device's $\overline{\text{SMBALERT}}$ line goes low, the following events occur:

- $\overline{\text{SMBALERT}}$ is pulled low.
- The master initiates a read operation and sends the alert response address (ARA = 0001 100). This is a general call address that must not be used as a specific device address.
- The device whose $\overline{\text{SMBALERT}}$ output is low responds to the alert response address, and the master reads its device address. The address of the device is now known and can be interrogated in the usual way.
- If more than one device's $\overline{\text{SMBALERT}}$ output is low, the one with the lowest device address has priority in accordance with normal SMBus arbitration.

Once the ADT7473/ADT7473−1 has responded to the alert response address, the master must read the status registers, and the $\overline{\text{SMBALERT}}$ is cleared only if the error condition is gone.

### SMBus Timeout

The ADT7473/ADT7473−1 includes an SMBus timeout feature. If there is no SMBus activity for 35 ms, the ADT7473/ADT7473−1 assumes the bus is locked and releases the bus. This prevents the device from locking or holding the SMBus expecting data. Some SMBus controllers cannot work with the SMBus timeout feature, so it can be disabled.

**Table 8. CONFIGURATION REGISTER 1 (REG. 0X40)**

| Bit | Description |
|---|---|
| <6> TODIS | 0: SMBus Timeout Enabled (Default)<br>1: SMBus Timeout Disabled |

### Voltage Measurement Input

The ADT7473/ADT7473−1 has one external voltage measurement channel and can also measure its own supply voltage, $V_{CC}$. Pin 14 can measure $V_{CCP}$. The $V_{CC}$ supply voltage measurement is carried out through the $V_{CC}$ pin (Pin 3). The $V_{CCP}$ input can be used to monitor a chipset supply voltage in computer systems.

### Analog-to-Digital Converter

All analog inputs are multiplexed into the on-chip, successive approximation, analog-to-digital converter. (ADC) This has a resolution of 10 bits. The basic input range is 0 V to 2.25 V, but the input has built-in attenuators to allow measurement of $V_{CCP}$ without any external components. To allow for the tolerance of the supply voltage, the ADC produces an output of 3/4 full scale (768 decimal or 300 hexadecimal) for the nominal input voltage and thus has adequate headroom to deal with overvoltages.

### Input Circuitry

The internal structure for the $V_{CCP}$ analog input is shown in Figure 23. The input circuit consists of an input protection diode, an attenuator, plus a capacitor to form a first order low-pass filter that provides the input immunity to high frequency noise.

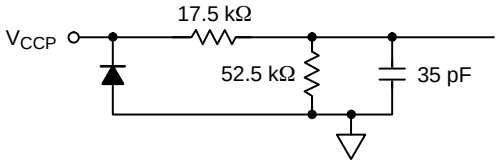



**Figure 23. Structure of Analog Inputs**

**Table 9. VOLTAGE MEASUREMENT REGISTERS**

| Register | Description | Default |
|---|---|---|
| 0x21 | $V_{CCP}$ Reading | 0x00 |
| 0x22 | $V_{CCP}$ Reading | 0x00 |

### $V_{CCP}$ Limit Registers

Associated with the $V_{CCP}$ measurement channel is a high and low limit register. Exceeding the programmed high or low limit causes the appropriate status bit to be set. Exceeding either limit can also generate $\overline{\text{SMBALERT}}$ interrupts.

**Table 10. $V_{CCP}$ LIMIT REGISTERS**

| Register | Description | Default |
|---|---|---|
| 0x46 | $V_{CCP}$ Low Limit | 0x00 |
| 0x47 | $V_{CCP}$ High Limit | 0xFF |

Table 13 shows the input ranges of the analog inputs and output codes of the 10-bit ADC.

When the ADC is running, it samples and converts a voltage input in 711 μs and averages 16 conversions to reduce noise; a measurement takes nominally 11.38 ms.

### Additional ADC Functions for Voltage Measurements

A number of other functions are available on the ADT7473/ADT7473−1 to offer the system designer increased flexibility.

### Turn-off Averaging

For each voltage measurement read from a value register, 16 readings have actually been made internally and the results averaged before being placed into the value register. When faster conversions are needed, setting Bit 4 of Configuration Register 2 (0x73) turns averaging off. This effectively gives a reading 16 times faster (711 μs), but the reading may be noisier.

### Bypass Voltage Input Attenuator

Setting Bit 5 of Configuration Register 2 (0x73) removes the attenuation circuitry from the $V_{CCP}$ input. This allows the user to directly connect external sensors or to rescale the analog voltage measurement inputs for other applications. The input range of the ADC without the attenuators is 0 V to 2.25 V.

### Single-channel ADC Conversion

Setting Bit 6 of Configuration Register 2 (0x73) places the ADT7473/ADT7473−1 into single-channel ADC conversion mode. In this mode, the ADT7473/ADT7473−1 can be made to read a single voltage channel only. If the internal ADT7473/ADT7473−1 clock is used, the selected input is read every 711 μs. The appropriate ADC channel is selected by writing to Bits <7:5> of the TACH1 minimum high byte register (0x55).

**Table 11. PROGRAMMING SINGLE-CHANNEL ADC MODE**

| Bits <7:5>, Register 0x55 | Channel Selected |
|---|---|
| 001 | $V_{CCP}$ |
| 010 | $V_{CC}$ |
| 101 | Remote 1 Temperature |
| 110 | Local Temperature |
| 111 | Remote 2 Temperature |

**Table 12. CONFIGURATION REGISTER 2 (REG. 0X73)**

| Bit | Description |
|---|---|
| <4> | 1: Averaging Off |
| <5> | 1: Bypass Input Attenuators |
| <6> | 1: Single-channel Convert Mode |

### TACH1 Minimum High Byte Register (0x55)

Bits <7:5> select ADC channel for single-channel convert mode.

**Table 13. 10-BIT ADC OUTPUT CODE VS. $V_{IN}$**

| ADC Output | | | |
|---|---|---|---|
| **$V_{CC}$ (3.3 $V_{IN}$)** (Note 1) | **$V_{CCP}$** | **Decimal** | **Binary (10 Bits)** |
| <0.0042 | <0.00293 | 0 | 00000000 00 |
| 0.0042 to 0.0085 | 0.0293 to 0.0058 | 1 | 00000000 01 |
| 0.0085 to 0.0128 | 0.0058 to 0.0087 | 2 | 00000000 10 |
| 0.0128 to 0.0171 | 0.0087 to 0.0117 | 3 | 00000000 11 |
| 0.0171 to 0.0214 | 0.0117 to 0.0146 | 4 | 00000001 00 |
| 0.0214 to 0.0257 | 0.0146 to 0.0175 | 5 | 00000001 01 |
| 0.0257 to 0.0300 | 0.0175 to 0.0205 | 6 | 00000001 10 |
| 0.0300 to 0.0343 | 0.0205 to 0.0234 | 7 | 00000001 11 |
| 0.0343 to 0.0386 | 0.0234 to 0.0263 | 8 | 00000010 00 |
| – | – | – | – |
| 1.100 to 1.1042 | 0.7500 to 0.7529 | 256 (1/4 scale) | 01000000 00 |
| – | – | – | – |
| 2.200 to 2.2042 | 1.5000 to 1.5029 | 512 (1/2 scale) | 10000000 00 |
| – | – | – | – |
| 3.300 to 3.3042 | 2.2500 to 2.2529 | 768 (3/4 scale) | 11000000 00 |
| – | – | – | – |
| 4.3527 to 4.3570 | 2.9677 to 2.9707 | 1013 | 11111101 01 |
| 4.3570 to 4.3613 | 2.9707 to 2.9736 | 1014 | 11111101 10 |
| 4.3613 to 4.3656 | 2.9736 to 2.9765 | 1015 | 11111101 11 |
| 4.3656 to 4.3699 | 2.9765 to 2.9794 | 1016 | 11111110 00 |
| 4.3699 to 4.3742 | 2.9794 to 2.9824 | 1017 | 11111110 01 |
| 4.3742 to 4.3785 | 2.9824 to 2.9853 | 1018 | 11111110 10 |
| 4.3785 to 4.3828 | 2.9853 to 2.9882 | 1019 | 11111110 11 |
| 4.3828 to 4.3871 | 2.9882 to 2.9912 | 1020 | 11111111 00 |
| 4.3871 to 4.3914 | 2.9912 to 2.9941 | 1021 | 11111111 01 |
| 4.3914 to 4.3957 | 2.9941 to 2.9970 | 1022 | 11111111 10 |
| >4.3957 | >2.9970 | 1023 | 11111111 11 |

1. The $V_{CC}$ output codes listed assume that $V_{CC}$ is 3.3 V.

### Temperature Measurement Method

A simple method of measuring temperature is to exploit the negative temperature coefficient of a diode, measuring the base-emitter voltage ($V_{BE}$) of a transistor operated at constant current. Unfortunately, this technique requires calibration to null out the effect of the absolute value of $V_{BE}$, which varies from device to device.

The technique used in the ADT7473/ADT7473−1 measures the change in $V_{BE}$ when the device is operated at three different currents. Previous devices have used only two operating currents, but the use of a third current allows automatic cancellation of resistances in series with the external temperature sensor.

Figure 24 shows the input signal conditioning used to measure the output of an external temperature sensor. This figure shows the external sensor as a substrate transistor, but it could equally be a discrete transistor. If a discrete transistor is used, the collector is not grounded and should be linked to the base. To prevent ground noise from interfering with the measurement, the more negative terminal of the sensor is not referenced to ground, but is biased above ground by an internal diode at the D− input. C1 can optionally be added as a noise filter (recommended maximum value 1,000 pF). However, a better option in noisy environments is to add a filter, as described in the Noise Filtering section.

### Local Temperature Measurement

The ADT7473/ADT7473−1 contains an on-chip band gap temperature sensor whose output is digitized by the on-chip 10-bit ADC. The 8-bit MSB temperature data is stored in the local temperature register (0x26). Because both positive and negative temperatures can be measured, the temperature data is stored in Offset 64 format or twos complement format, as shown in Table 14 and Table 15. Theoretically, the temperature sensor and ADC can measure temperatures from −63°C to +127°C (or −63°C to +191°C in the extended temperature range) with a resolution of +0.25°C. However, this exceeds the operating temperature range of the device, so local temperature measurements outside the ADT7473/ADT7473−1 operating temperature range are not possible.

### Remote Temperature Measurement

The ADT7473/ADT7473−1 can measure the temperature of two remote diode sensors or diode-connected transistors connected to Pin 10 and Pin 11 or to Pin 12 and Pin 13.

The forward voltage of a diode or diode-connected transistor operated at a constant current exhibits a negative temperature coefficient of about −2 mV/°C. Unfortunately, the absolute value of $V_{BE}$ varies from device to device and individual calibration is required to null this out, so the technique is unsuitable for mass production. The technique used in the ADT7473/ADT7473−1 is to measure the change in $V_{BE}$ when the device is operated at three different currents. This is given by:

$$\Delta V_{BE} = kT/q \times \ln(N) \quad \text{(eq. 1)}$$

where:
k is Boltzmann's constant.
T is the absolute temperature in Kelvin.
q is the charge on the carrier.
N is the ratio of the two currents.

Figure 24 shows the input signal conditioning used to measure the output of a remote temperature sensor. This figure shows the external sensor as a substrate transistor, provided for temperature monitoring on some microprocessors. It could also be a discrete transistor such as a 2N3904/2N3906.

**Table 14. TWOS COMPLEMENT TEMPERATURE DATA FORMAT**

| Temperature | **Digital Output (10-bit)** (Note 1) |
|---|---|
| –128°C | 1000 0000 **00** (Diode Fault) |
| –63°C | 1100 0001 **00** |
| –50°C | 1100 1110 **00** |
| –25°C | 1110 0111 **00** |
| –10°C | 1111 0110 **00** |
| 0°C | 0000 0000 **00** |
| 10.25°C | 0000 1010 **01** |
| 25.5°C | 0001 1001 **10** |
| 50.75°C | 0011 0010 **11** |
| 75°C | 0100 1011 **00** |
| 100°C | 0110 0100 **00** |
| 125°C | 0111 1101 **00** |
| 127°C | 0111 1111 **00** |

1. Bold numbers denote 2 LSBs of measurement in the Extended Resolution Register 2 (Register 0x77) with 0.25°C resolution.

**Table 15. EXTENDED RANGE, TEMPERATURE DATA FORMAT**

| Temperature | **Digital Output (10-bit)** (Note 1) |
|---|---|
| –64°C | 0000 0000 **00** (Diode Fault) |
| –63°C | 0000 0001 **00** |
| –1°C | 0011 1111 **00** |
| 0°C | 0100 0000 **00** |
| 1°C | 0100 0001 **00** |
| 10°C | 0100 1010 **00** |
| 25°C | 0101 1001 **00** |
| 50°C | 0111 0010 **00** |
| 75°C | 1000 1001 **00** |
| 100°C | 1010 0100 **00** |
| 125°C | 1011 1101 **00** |
| 191°C | 1111 1111 **00** |

1. Bold numbers denote 2 LSBs of measurement in the Extended Resolution Register 2 (Register 0x77) with 0.25°C resolution.

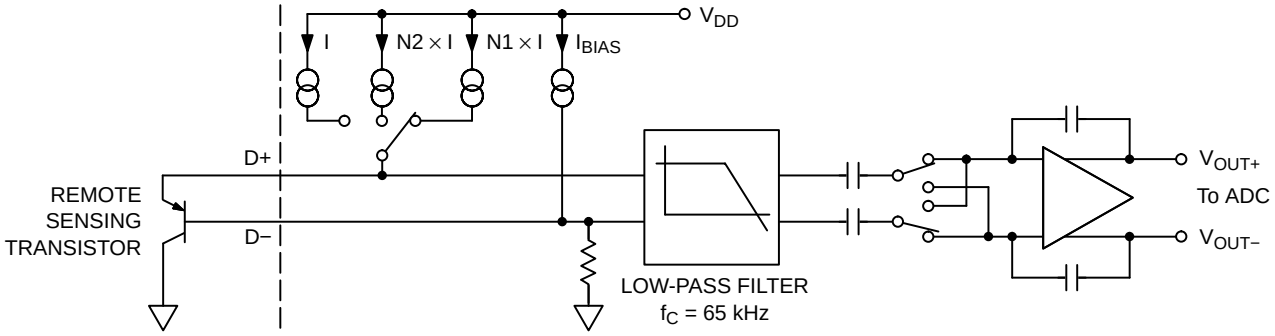



**Figure 24. Signal Conditioning for Remote Diode Temperature Sensors**

If a discrete transistor is used, the collector is not grounded and should be linked to the base. If a PNP transistor is used, the base is connected to the D– input and the emitter is connected to the D+ input. If an NPN transistor is used, the emitter is connected to the D– input and the base is connected to the D+ input. Figure 25 and Figure 26 show how to connect the ADT7473/ADT7473−1 to an NPN or PNP transistor for temperature measurement. To prevent ground noise from interfering with the measurement, the more negative terminal of the sensor is not referenced to ground, but is biased above ground by an internal diode at the D– input.

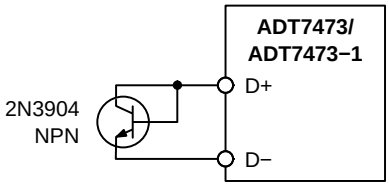



**Figure 25. Measuring Temperature by Using an NPN Transistor**

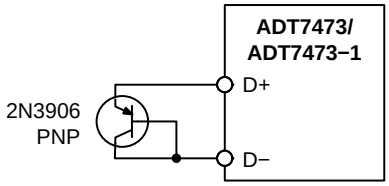



**Figure 26. Measuring Temperature by Using a PNP Transistor**

To measure $\Delta V_{BE}$, the operating current through the sensor is switched among three related currents. N1 × I and N2 × I are different multiples of the current I, as shown in Figure 24. The currents through the temperature diode are switched between I and N1 × I, giving $\Delta V_{BE1}$, and then between I and N2 × I, giving $\Delta V_{BE2}$. The temperature can then be calculated using the two $\Delta V_{BE}$ measurements. This method can also cancel the effect of any series resistance on the temperature measurement.

The resulting $\Delta V_{BE}$ waveforms are passed through a 65 kHz low-pass filter to remove noise and then to a chopper-stabilized amplifier. This amplifies and rectifies the waveform to produce a dc voltage proportional to $\Delta V_{BE}$. The ADC digitizes this voltage, and a temperature measurement is produced. To reduce the effects of noise, digital filtering is performed by averaging the results of 16 measurement cycles.

The results of remote temperature measurements are stored in 10-bit, twos complement format, as listed in Table 10. The extra resolution for the temperature measurements is held in the Extended Resolution Register 2 (0x77). This gives temperature readings with a resolution of 0.25°C.

### Noise Filtering

For temperature sensors operating in noisy environments, previous practice was to place a capacitor across the D+ pin and the D− pin to help combat the effects of noise. However, large capacitances affect the accuracy of the temperature measurement, leading to a recommended maximum capacitor value of 1,000 pF. This capacitor reduces the noise, but does not eliminate it, making use of the sensor difficult in a very noisy environment.

The ADT7473/ADT7473−1 has a major advantage over other devices for eliminating the effects of noise on the external sensor. Using the series resistance cancellation feature, a filter can be constructed between the external temperature sensor and the part. The effect of any filter resistance seen in series with the remote sensor is automatically canceled from the temperature result.

The construction of a filter allows the ADT7473/ADT7473−1 and the remote temperature sensor to operate in noisy environments. Figure 27 shows a low-pass R-C filter with the following values:

$$R = 100\ \Omega,\ C = 1\ nF \qquad \text{(eq. 1)}$$

This filtering reduces both common-mode noise and differential noise.

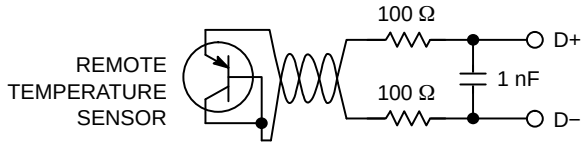



**Figure 27. Filter Between Remote Sensor and ADT7473/ADT7473−1**

### Series Resistance Cancellation

Parasitic resistance to the ADT7473/ADT7473−1 D+ and D− inputs (seen in series with the remote diode) is caused by a variety of factors including PCB track resistance and track length. This series resistance appears as a temperature offset in the remote sensor's temperature measurement. This error typically causes a 0.5°C offset per Ω of parasitic resistance in series with the remote diode.

The ADT7473/ADT7473−1 automatically cancels out the effect of this series resistance on the temperature reading, giving a more accurate result without the need for user characterization of this resistance. The ADT7473/ADT7473−1 is designed to automatically cancel up to 3 kΩ of resistance, typically. This is transparent to the user by using an advanced temperature measurement method. This feature allows resistances to be added to the sensor path to produce a filter, allowing the part to be used in noisy environments. See the Noise Filtering section for details.

## Factors Affecting Diode Accuracy

### Remote Sensing Diode

The ADT7473/ADT7473−1 is designed to work with either substrate transistors built into processors or discrete transistors. Substrate transistors are generally PNP types with the collector connected to the substrate. Discrete types can be either PNP or NPN transistors connected as a diode (base-shorted to the collector). If an NPN transistor is used, the collector and base are connected to D+ and the emitter is connected to D−. If a PNP transistor is used, the collector and base are connected to D− and the emitter is connected to D+.

To reduce the error due to variations in both substrate and discrete transistors, a number of factors should be taken into consideration:

- The ideality factor, $n_f$, of the transistor is a measure of the deviation of the thermal diode from ideal behavior. The ADT7473/ADT7473−1 is trimmed for an $n_f$ value of 1.008. Use the following equation to calculate the error introduced at a temperature, T(°C), when using a transistor whose $n_f$ does not equal 1.008. Refer to the data sheet for the related CPU to obtain the $n_f$ values.

$$\Delta T = (n_f - 1.008)/1.008 \times (273.15\ K + T) \quad \text{(eq. 2)}$$

  To factor this in, the user can write the ΔT value to the offset register. Then, the ADT7473/ADT7473−1 automatically adds it to or subtracts it from the temperature measurement.
- Some CPU manufacturers specify the high and low current levels of the substrate transistors. The high current level of the ADT7473/ADT7473−1, $I_{HIGH}$, is 96 μA and the low level current, $I_{LOW}$, is 6 μA. If the ADT7473/ADT7473−1 current levels do not match the current levels specified by the CPU manufacturer, it might be necessary to remove an offset. The CPU's data sheet advises whether this offset needs to be removed and how to calculate it. This offset can be programmed to the offset register. It is important to note that, if more than one offset must be considered, the algebraic sum of these offsets must be programmed to the offset register.

If a discrete transistor is used with the ADT7473/ADT7473−1, the best accuracy is obtained by choosing devices according to the following criteria:

- Base-emitter voltage greater than 0.25 V at 6 μA, at the highest operating temperature.
- Base-emitter voltage less than 0.95 V at 100 μA, at the lowest operating temperature.
- Base resistance less than 100 Ω.
- Small variation in $h_{FE}$ (such as 50 to 150) that indicates tight control of $V_{BE}$ characteristics.

Transistors, such as 2N3904, 2N3906, or equivalents in SOT−23 packages, are suitable devices to use.

### Nulling Out Temperature Errors

As CPUs run faster, it becomes more difficult to avoid high frequency clocks when routing the D+/D– traces around a system board. Even when recommended layout guidelines are followed, some temperature errors can still be attributable to noise coupled onto the D+/D– lines. Constant high frequency noise usually attenuates or increases temperature measurements by a linear, constant value.

The ADT7473/ADT7473−1 has temperature offset registers at Register 0x70 and Register 0x72 for the Remote 1 and Remote 2 temperature channels. By performing a one-time calibration of the system, the user can determine the offset caused by system board noise and null it out using the offset registers. The offset registers automatically add a twos complement, 8-bit reading to every temperature measurement. The LSBs add +0.5°C offset to the temperature reading so the 8-bit register effectively allows temperature offsets of up to ±64°C with a resolution of +0.5°C. This ensures that the readings in the temperature measurement registers are as accurate as possible.

**Table 16. TEMPERATURE OFFSET REGISTERS**

| Register | Description | Default |
|---|---|---|
| 0x70 | Remote 1 Temperature Offset | 0x00 (0°C) |
| 0x71 | Local Temperature Offset | 0x00 (0°C) |
| 0x72 | Remote 2 Temperature Offset | 0x00 (0°C) |

### ADT7460/ADT7473/ADT7473−1 Backwards-compatible Mode

By setting Bit 1 of Configuration Register 5 (0x7C), all temperature measurements are stored in the zone temperature value registers (Register 0x25, Register 0x26, and Register 0x27) in twos complement, in the range −63°C to +127°C. (The ADT7473/ADT7473−1 still makes calculations based on the Offset 64 extended range and clamps the results, if necessary.) The temperature limits

must be reprogrammed in twos complement. If a twos complement temperature below −63°C is entered, the temperature is clamped to −63°C. In this mode, the diode fault condition remains −128°C = 1000 0000, while in the extended temperature range (−64°C to +191°C), the fault condition is represented by −64°C = 0000 0000.

**Table 17. TEMPERATURE MEASUREMENT REGISTERS**

| Register | Description | Default |
|---|---|---|
| 0x25 | Remote 1 Temperature | − |
| 0x26 | Local Temperature | − |
| 0x27 | Remote 2 Temperature | − |
| 0x77 | Extended Resolution 2 | 0x00 |

**Table 18. EXTENDED RESOLUTION TEMPERATURE MEASUREMENT REGISTER BITS**

| Bit | Mnemonic | Description |
|---|---|---|
| <7:6> | TDM2 | Remote 2 Temperature LSBs |
| <5:4> | LTMP | Local Temperature LSBs |
| <3:2> | TDM1 | Remote 1 Temperature LSBs |

### Temperature Measurement Limit Registers

Associated with each temperature measurement channel are high and low limit registers. Exceeding the programmed high or low limit causes the appropriate status bit to be set. Exceeding either limit can also generate $\overline{\text{SMBALERT}}$ interrupts.

**Table 19. TEMPERATURE MEASUREMENT LIMIT REGISTERS**

| Register | Description | Default |
|---|---|---|
| 0x4E | Remote 1 Temperature Low Limit | 0x01 |
| 0x4F | Remote 1 Temperature High Limit | 0x7F |
| 0x50 | Local Temperature Low Limit | 0x01 |
| 0x51 | Local Temperature High Limit | 0x7F |
| 0x52 | Remote 2 Temperature Low Limit | 0x01 |
| 0x53 | Remote 2 Temperature High Limit | 0x7F |

### Reading Temperature from the ADT7473/ADT7473−1

It is important to note that the temperature can be read from the ADT7473/ADT7473−1 as an 8-bit value (with 1°C resolution) or as a 10-bit value (with 0.25°C resolution). If only 1°C resolution is required, the temperature readings can be read back at any time and in no particular order.

If the 10-bit measurement is required, a 2-register read for each measurement is used. The extended resolution register (Register 0x77) should be read first. This causes all temperature reading registers to be frozen until all temperature reading registers have been read from. This prevents an MSB reading from being updated while its two LSBs are being read, and vice versa.

### Additional ADC Functions for Temperature Measurement

A number of other functions are available on the ADT7473/ADT7473−1 to offer the system designer increased flexibility.

### Turn-off Averaging

For each temperature measurement read from a value register, 16 readings have actually been made internally and the results averaged before being placed into the value register. Sometimes it is necessary to take a very fast measurement. Setting Bit 4 of Configuration Register 2 (0x73) turns averaging off.

**Table 20. CONVERSION TIME WITH AVERAGING DISABLED**

| Channel | Measurement Time (ms) |
|---|---|
| Voltage Channel | 0.7 |
| Remote 1 Temperature | 7 |
| Remote 2 Temperature | 7 |
| Local Temperature | 1.3 |

**Table 21. CONVERSION TIME WITH AVERAGING ENABLED**

| Channel | Measurement Time (ms) |
|---|---|
| Voltage Channel | 11 |
| Remote Temperature | 39 |
| Local Temperature | 12 |

### Single-channel ADC Conversions

Setting Bit 6 of Configuration Register 2 (0x73) places the ADT7473/ADT7473−1 into single-channel ADC conversion mode. In this mode, the ADT7473/ADT7473−1 can be made to read a single temperature channel only. The appropriate ADC channel is selected by writing to Bits <7:5> of the TACH1 minimum high byte register (0x55).

**Table 22. PROGRAMMING SINGLE-CHANNEL ADC MODE FOR TEMPERATURES**

| Channel Selected | Bits <7:4>, Register 0x55 |
|---|---|
| 101 | Remote 1 Temperature |
| 110 | Local Temperature |
| 111 | Remote 2 Temperature |

### Configuration Register 2 (0x73)

Bit <4> = 1, Averaging Off.

Bit <6> = 1, Single-channel Convert Mode.

### TACH1 Minimum High Byte Register (0x55)

Bits <7:5> select the ADC channel for single-channel convert mode.

### Overtemperature Events

Overtemperature events on any of the temperature channels can be detected and dealt with automatically in

automatic fan speed control mode. Register 0x6A to Register 0x6C are the $\overline{\text{THERM}}$ limits. When a temperature exceeds its $\overline{\text{THERM}}$ limit, all PWM outputs run at 100% duty cycle or the maximum PWM duty cycle (Register 0x38, Register 0x39, and Register 0x3A) if Bit 3 of Configuration Register 4 (0x7D) is set. The fans remain running at this speed until the temperature drops below $\overline{\text{THERM}}$ minus hysteresis; this can be disabled by setting the boost bit in Configuration Register 3 (0x78), Bit 2. The hysteresis value for that $\overline{\text{THERM}}$ limit is the value programmed into the hysteresis registers (Register 0x6D and Register 0x6E). The default hysteresis value is 4°C.

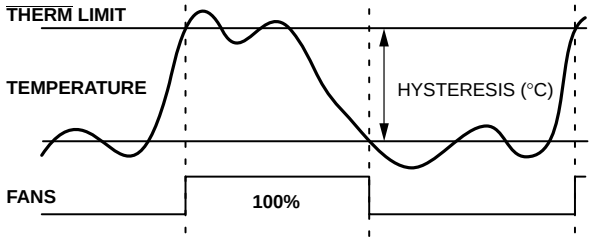



**Figure 28. $\overline{\text{THERM}}$ Limit Operation**

## Limits, Status Registers, and Interrupts

### Limit Values

Associated with each measurement channel on the ADT7473/ADT7473−1 are high and low limits. These can form the basis of system status monitoring; a status bit can be set for any out-of-limit condition and is detected by polling the device. Alternatively, $\overline{\text{SMBALERT}}$ interrupts can be generated to flag a processor or microcontroller of out-of-limit conditions.

### 8-bit Limits

The following is a list of 8-bit limits on the ADT7473/ADT7473−1.

**Table 23. VOLTAGE LIMIT REGISTERS**

| Register | Description | Default |
|---|---|---|
| 0x46 | $V_{CCP}$ Low Limit | 0x00 |
| 0x47 | $V_{CCP}$ High Limit | 0xFF |
| 0x48 | $V_{CC}$ Low Limit | 0x00 |
| 0x49 | $V_{CC}$ High Limit | 0xFF |

**Table 24. TEMPERATURE LIMIT REGISTERS**

| Register | Description | Default |
|---|---|---|
| 0x4E | Remote 1 Temperature Low Limit | 0x01 |
| 0x4F | Remote 1 Temperature High Limit | 0xFF |
| 0x6A | Remote 1 $\overline{\text{THERM}}$ Limit | 0xA4 |
| 0x50 | Local Temperature Low Limit | 0x01 |
| 0x51 | Local Temperature High Limit | 0xFF |
| 0x6B | Local $\overline{\text{THERM}}$ Temperature Limit | 0xA4 |
| 0x52 | Remote 2 Temperature Low Limit | 0x01 |
| 0x53 | Remote 2 Temperature High Limit | 0xFF |
| 0x6C | Remote 2 $\overline{\text{THERM}}$ Temp. Limit | 0xA4 |

**Table 25. $\overline{\text{THERM}}$ LIMIT REGISTER**

| Register | Description | Default |
|---|---|---|
| 0x7A | $\overline{\text{THERM}}$ Timer Limit | 0x00 |

### 16-bit Limits

The fan TACH measurements are 16-bit results. The fan TACH limits are also 16 bits, consisting of a high byte and low byte. Because fans running under speed or stalled are normally the only conditions of interest, only high limits exist for fan TACHs. Because the fan TACH period is actually being measured, exceeding the limit indicates a slow or stalled fan.

**Table 26. FAN LIMIT REGISTERS**

| Register | Description | Default |
|---|---|---|
| 0x54 | TACH1 Minimum Low Byte | 0xFF |
| 0x55 | TACH1 Minimum High Byte | 0xFF |
| 0x56 | TACH2 Minimum Low Byte | 0xFF |
| 0x57 | TACH2 Minimum High Byte | 0xFF |
| 0x58 | TACH3 Minimum Low Byte | 0xFF |
| 0x59 | TACH3 Minimum High Byte | 0xFF |
| 0x5A | TACH4 Minimum Low Byte | 0xFF |
| 0x5B | TACH4 Minimum High Byte | 0xFF |

### Out-of-Limit Comparisons

Once all limits have been programmed, the ADT7473/ADT7473−1 can be enabled for monitoring. The ADT7473/ADT7473−1 measures all voltage and temperature measurements in round-robin format and sets the appropriate status bit for out-of-limit conditions. TACH measurements are not part of this round-robin cycle. Comparisons are done differently depending on whether the measured value is being compared to a high or low limit.

High limit > comparison performed
Low limit ≤ comparison performed

Voltage and temperature channels use a window comparator for error detecting and, therefore, have high and low limits. Fan speed measurements use only a low limit. This fan limit is needed only in manual fan control mode.

### Analog Monitoring Cycle Time

The analog monitoring cycle begins when a 1 is written to the start bit (Bit 0) of Configuration Register 1 (0x40). By default, the ADT7473/ADT7473−1 powers up with this bit set. The ADC measures each analog input in turn and, as each measurement is completed, the result is automatically stored in the appropriate value register. This round-robin monitoring cycle continues unless disabled by writing a 0 to Bit 0 of Configuration Register 1.

As the ADC is normally left to free-run in this manner, the time taken to monitor all the analog inputs is normally not of interest, because the most recently measured value of any input can be read out at any time.

For applications where the monitoring cycle time is important, it can easily be calculated. The total number of channels measured is:

- One Dedicated Supply Voltage Input ($V_{CCP}$)
- Supply Voltage ($V_{CC}$ Pin)
- Local Temperature
- Two Remote Temperatures

As mentioned previously, the ADC performs round-robin conversions. The total monitoring cycle time for averaged voltage and temperature monitoring is 146 ms. The total monitoring cycle time for voltage and temperature monitoring with averaging disabled is 19 ms. The ADT7473/ ADT7473−1 is a derivative of the ADT7467. As a result, the total conversion time in the ADT7473/ ADT7473−1 is the same as the total conversion time of the ADT7467, even though the ADT7473/ADT7473−1 has fewer monitored channels.

Fan TACH measurements are made in parallel and are not synchronized with the analog measurements in any way.

### Interrupt Status Registers

The results of limit comparisons are stored in Interrupt Status Register 1 and Interrupt Status Register 2. The status register bit for each channel reflects the status of the last measurement and limit comparison on that channel. If a measurement is within limits, the corresponding status register bit is cleared to 0. If the measurement is out of limits, the corresponding status register bit is set to 1.

The state of the various measurement channels can be polled by reading the status registers over the serial bus. In Bit 7 (OOL) of Interrupt Status Register 1 (Reg. 0x41), a 1 means an out-of-limit event has been flagged in Interrupt Status Register 2. This means the user needs only to read Interrupt Status Register 2 when this bit is set. Alternatively, Pin 5 or Pin 9 on the ADT7473 can be configured as an $\overline{\text{SMBALERT}}$ output, while only Pin 9 can be configured to be an $\overline{\text{SMBALERT}}$ on the ADT7473−1. This automatically notifies the system supervisor of an out-of-limit condition. Reading the status registers clears the appropriate status bit as long as the error condition that caused the interrupt has cleared. Status register bits (except OVT) are sticky. Whenever a status bit is set, indicating an out-of-limit condition, it remains set even if the event that caused it has gone away (until read). The only way to clear the status bit is to read the status register after the event has gone away. Interrupt mask registers (Register 0x74 and Register 0x75) allow individual interrupt sources to be masked from causing an $\overline{\text{SMBALERT}}$. However, if one of these masked interrupt sources goes out of limit, its associated status bit is set in the interrupt status registers. OVT clears automatically.

**Table 27. INTERRUPT STATUS REGISTER 1 (REG. 0X41)**

| Bit | Mnemonic | Description |
|---|---|---|
| 7 | OOL | 1 denotes a bit in Interrupt Status Register 2 is set and Interrupt Status Register 2 should be read. |
| 6 | R2T | 1 indicates that the Remote 2 Temperature High or Low Limit has been exceeded. |
| 5 | LT | 1 indicates that the Local Temperature High or Low Limit has been exceeded. |
| 4 | R1T | 1 indicates that the Remote 1 Temperature High or Low Limit has been exceeded. |
| 3 | – | Unused |
| 2 | $V_{CC}$ | 1 indicates that the $V_{CC}$ High or Low Limit has been exceeded. |
| 1 | $V_{CCP}$ | 1 indicates that the $V_{CCP}$ High or Low Limit has been exceeded. |
| 0 | – | Unused |

**Table 28. INTERRUPT STATUS REGISTER 2 (REG. 0X42)**

| Bit | Mnemonic | Description |
|---|---|---|
| 7 | D2 | 1 indicates an open or short on D2+/D2− inputs. |
| 6 | D1 | 1 indicates an open or short on D1+/D1− inputs. |
| 5 | F4P | 1 indicates that Fan 4 has dropped below the minimum speed. Alternatively, indicates that $\overline{\text{THERM}}$ timer limit has been exceeded if the $\overline{\text{THERM}}$ timer function is used. |
| 4 | FAN3 | 1 indicates that Fan 3 has dropped below the minimum speed. |
| 3 | FAN2 | 1 indicates that Fan 2 has dropped below the minimum speed. |
| 2 | FAN1 | 1 indicates that Fan 1 has dropped below the minimum speed. |
| 1 | OVT | 1 indicates that a $\overline{\text{THERM}}$ overtemperature limit has been exceeded. |
| 0 | $\overline{\text{THERM}}$ Limit Latch | 1 indicates that a Remote Channel 2 latch. |

### $\overline{\text{SMBALERT}}$ Interrupt Behavior

The ADT747/ADT7473−1 can be polled for status, or an $\overline{\text{SMBALERT}}$ interrupt can be generated for out-of-limit conditions. It is important to note how the $\overline{\text{SMBALERT}}$ output and status bits behave when writing interrupt handler software.

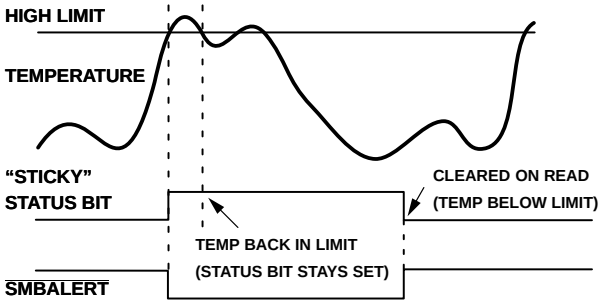



**Figure 29. $\overline{\text{SMBALERT}}$ and Status Bit Behavior**

Figure 29 shows how the $\overline{\text{SMBALERT}}$ output and sticky status bits behave. Once a limit is exceeded, the corresponding status bit is set to 1. The interrupt status bit remains set until the error condition subsides and the interrupt status register is read. The status bits are referred to as sticky because they remain set until read by software. This ensures that an out-of-limit event cannot be missed if software is polling the device periodically. Note that the $\overline{\text{SMBALERT}}$ output remains low for the entire duration that a reading is out of limit and until the interrupt status register has been read. This has implications on how software handles the interrupt.

Note that $\overline{\text{THERM}}$ overtemperature events are not sticky, resetting immediately after the overtemperature condition ceases. This also applies to $\overline{\text{SMBALERT}}$ if associated with an OVT event.

### Handling $\overline{\text{SMBALERT}}$ Interrupts

To prevent the system from being tied up servicing interrupts, it is recommended to handle the $\overline{\text{SMBALERT}}$ interrupt as follows:

1. Detect the $\overline{\text{SMBALERT}}$ assertion.
2. Enter the interrupt handler.
3. Read the status registers to identify the interrupt source.
4. Mask the interrupt source by setting the appropriate mask bit in the interrupt mask registers (Register 0x74 and Register 0x75).
5. Take the appropriate action for a given interrupt source.
6. Exit the interrupt handler.

Periodically poll the status registers. If the interrupt status bit has cleared, reset the corresponding interrupt mask bit to 0. This causes the $\overline{\text{SMBALERT}}$ output and status bits to behave as shown in Figure 30.

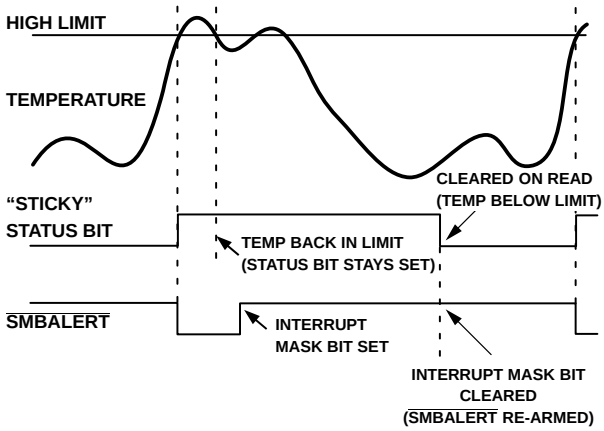



**Figure 30. How Masking the Interrupt Source Affects $\overline{\text{SMBALERT}}$ Output**

### Masking Interrupt Sources

Register 0x74, Interrupt Mask Register 1

Register 0x75, Interrupt Mask Register 2

These registers allow individual interrupt sources to be masked out to prevent $\overline{\text{SMBALERT}}$ interrupts. Masking an interrupt source prevents only the $\overline{\text{SMBALERT}}$ output from being asserted; the appropriate status bit is set normally.

**Table 29. INTERRUPT MASK REGISTER 1 (REG. 0X74)**

| Bit | Mnemonic | Description |
|---|---|---|
| 7 | OOL | 0 when one or more alerts are generated in Interrupt Status Register 2, assuming all the mask bits in the Interrupt Mask Register 2 (0x75) = 1; $\overline{\text{SMBALERT}}$ is still asserted.<br>1 when one or more alerts are generated in Interrupt Status Register 2, assuming all the mask bits in the Interrupt Mask Register 2 (0x75) = 1; $\overline{\text{SMBALERT}}$ is not asserted. |
| 6 | R2T | 1 masks $\overline{\text{SMBALERT}}$ for Remote 2 temperature. |
| 5 | LT | 1 masks $\overline{\text{SMBALERT}}$ for Local temperature. |
| 4 | R1T | 1 masks $\overline{\text{SMBALERT}}$ for Remote 1 temperature. |
| 3 | − | Unused |
| 2 | $V_{CC}$ | 1 masks $\overline{\text{SMBALERT}}$ for the $V_{CC}$ channel. |
| 1 | $V_{CCP}$ | 1 masks $\overline{\text{SMBALERT}}$ for the $V_{CCP}$ channel. |
| 0 | − | Unused |

**Table 30. INTERRUPT MASK REGISTER 2 (REG. 0X75)**

| Bit | Mnemonic | Description |
|---|---|---|
| 7 | D2 | 1 masks $\overline{\text{SMBALERT}}$ for Diode 2 errors. |
| 6 | D1 | 1 masks $\overline{\text{SMBALERT}}$ for Diode 1 errors. |
| 5 | FAN4 | 1 masks $\overline{\text{SMBALERT}}$ for Fan 4 failure. If the TACH4 pin is being used as the $\overline{\text{THERM}}$ input, this bit masks $\overline{\text{SMBALERT}}$ for a $\overline{\text{THERM}}$ event. |
| 4 | FAN3 | 1 masks $\overline{\text{SMBALERT}}$ for Fan 3. |
| 3 | FAN2 | 1 masks $\overline{\text{SMBALERT}}$ for Fan 2. |
| 2 | FAN1 | 1 masks $\overline{\text{SMBALERT}}$ for Fan 1. |
| 1 | OVT | 1 masks $\overline{\text{SMBALERT}}$ for overtemperature (exceeding $\overline{\text{THERM}}$ limits). |
| 0 | − | Unused |

### Enabling the $\overline{\text{SMBALERT}}$ Interrupt Output

The $\overline{\text{SMBALERT}}$ interrupt function is disabled by default. Pin 5 or Pin 9 can be reconfigured as an $\overline{\text{SMBALERT}}$ output to signal out-of-limit conditions. ($\overline{\text{SMBALERT}}$ function is available only on Pin 9 of ADT7473−1.)

**Table 31. ADT7473 CONFIGURING PIN 5 AS $\overline{\text{SMBALERT}}$ OUTPUT**

| Register | Bit Setting |
|---|---|
| Configuration Register 3 (Register 0x78) | <0> $\overline{\text{ALERT}}$ = 1 |

The ADT7473−1 $\overline{\text{THERM\_LATCH}}$ function latches and asserts when temperature rises 0.25°C above the $\overline{\text{THERM}}$ limit for the selected remote channel. Due to a $\overline{\text{THERM}}$ event, the fans spin at full speed. This can be disabled by setting Bit 2 in Configuration Register 0x7D.

Pin 5 remains latched until temperature falls below $\overline{\text{THERM}}$ limit for the selected zone, Remote Channel D1 or Remote Channel D2, and Bit 0 in Status Register 2 is cleared. By default on the ADT7473−1, the $\overline{\text{THERM}}$ limit is set as 136°C for Remote Channel 2 and 100°C for Remote Channel 1.

### Assigning $\overline{\text{THERM}}$ Functionality to a Pin

Pin 9 on the ADT7473/ADT7473−1 has four possible functions: $\overline{\text{SMBALERT}}$, $\overline{\text{THERM}}$, GPIO, and TACH4. The user chooses the required functionality by setting Bit 0 and Bit 1 of Configuration Register 4 (0x7D).

**Table 32. CONFIGURATION REGISTER 4 (REG. 0X7D)**

| Bit 1 | Bit 0 | Function |
|---|---|---|
| 0 | 1 | TACH4 |
| 0 | 0 | $\overline{\text{THERM}}$ |
| 1 | 1 | SMBusALERT |
| 1 | 0 | GPIO |

Once Pin 9 is configured as $\overline{\text{THERM}}$, it must be enabled by setting Bit 1 of Configuration Register 3 (0x78).

### $\overline{\text{THERM}}$ as an Input

When $\overline{\text{THERM}}$ is configured as an input, the ADT7473/ADT7473−1 can time assertions on the $\overline{\text{THERM}}$ pin. This can be useful for connecting to the $\overline{\text{PROCHOT}}$ output of a CPU to gauge system performance. See the $\overline{\text{THERM}}$ Timer section for more information.

The user can also set up the ADT7473/ADT7473−1 so that, when the $\overline{\text{THERM}}$ pin is driven low externally, the fans run at 100%. The fans run at 100% for the duration of the time the $\overline{\text{THERM}}$ pin is pulled low. This is done by setting the BOOST bit (Bit 2) in Configuration Register 3 (0x78) to 1. This works only if the fan is already running, for example, in manual mode when the current duty cycle is above 0x00, or in automatic mode when the temperature is above $T_{MIN}$. If the temperature is below $T_{MIN}$ or if the duty cycle in manual mode is set to 0x00, then pulling the $\overline{\text{THERM}}$ low externally has no effect. See Figure 31 for more information.

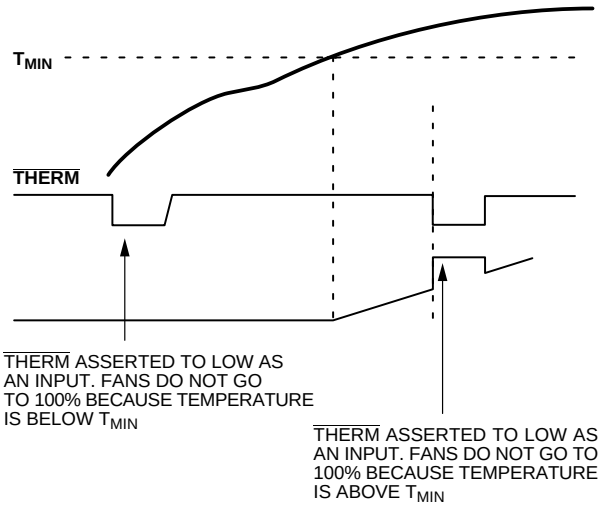



**Figure 31. Asserting $\overline{\text{THERM}}$ Low as an Input in Automatic Fan Speed Control Mode**

### $\overline{\text{THERM}}$ Timer

The ADT7473/ADT7473−1 has an internal timer to measure $\overline{\text{THERM}}$ assertion time. For example, the $\overline{\text{THERM}}$ input can be connected to the $\overline{\text{PROCHOT}}$ output of a Pentium® 4 CPU to measure system performance. The $\overline{\text{THERM}}$ input can also be connected to the output of a trip point temperature sensor.

The timer is started on the assertion of the ADT7473/ADT7473−1 $\overline{\text{THERM}}$ input and stopped when $\overline{\text{THERM}}$ is deasserted. The timer counts $\overline{\text{THERM}}$ times cumulatively; that is, the timer resumes counting on the next $\overline{\text{THERM}}$ assertion. The $\overline{\text{THERM}}$ timer continues to accumulate $\overline{\text{THERM}}$ assertion times until the timer is read (it is cleared on read) or until it reaches full scale. If the counter reaches full scale, it stops at that reading until cleared.

The 8-bit $\overline{\text{THERM}}$ timer status register (0x79) is designed so that Bit 0 is set to 1 on the first $\overline{\text{THERM}}$ assertion. Once the cumulative $\overline{\text{THERM}}$ assertion time has exceeded

45.52 ms, Bit 1 of the $\overline{\text{THERM}}$ timer is set and Bit 0 becomes the LSB of the timer with a resolution of 22.76 ms (see Figure 32).

When using the $\overline{\text{THERM}}$ timer, be aware of the following. After a $\overline{\text{THERM}}$ timer read (0x79):

1. The contents of the timer are cleared on read.
2. The F4P bit (Bit 5) of Interrupt Status Register 2 needs to be cleared (assuming that the $\overline{\text{THERM}}$ timer limit has been exceeded).

If the $\overline{\text{THERM}}$ timer is read during a $\overline{\text{THERM}}$ assertion, then the following happens:

1. The contents of the timer are cleared.
2. Bit 0 of the $\overline{\text{THERM}}$ timer is set to 1 (because a $\overline{\text{THERM}}$ assertion is occurring).
3. The $\overline{\text{THERM}}$ timer increments from 0.
4. If the $\overline{\text{THERM}}$ timer limit (Register 0x7A) = 0x00, the F4P bit is set.

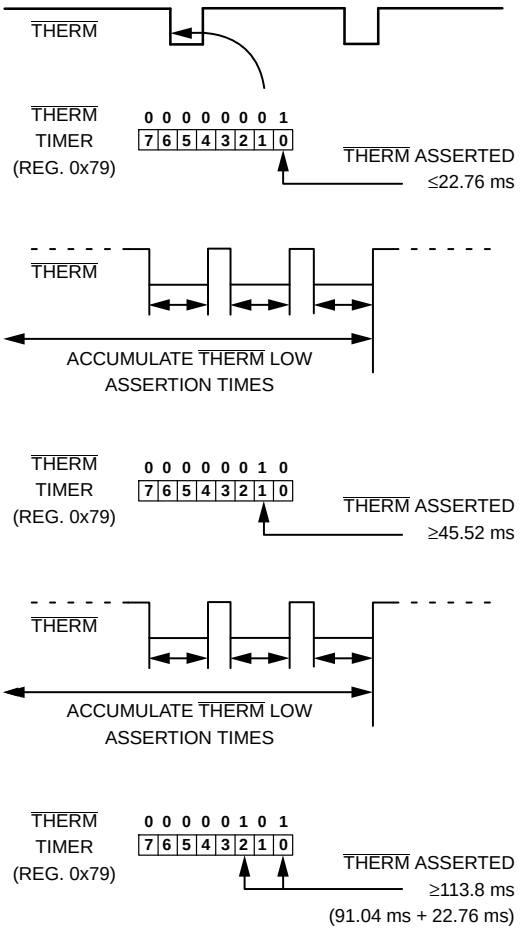



**Figure 32. Understanding the $\overline{\text{THERM}}$ Timer**

### Generating $\overline{\text{SMBALERT}}$ Interrupts from $\overline{\text{THERM}}$ Timer Events

The ADT7473/ADT7473−1 can generate an $\overline{\text{SMBALERT}}$ when a programmable $\overline{\text{THERM}}$ timer limit is exceeded. This allows the system designer to ignore brief, infrequent $\overline{\text{THERM}}$ assertions, while capturing longer $\overline{\text{THERM}}$ timer events. Register 0x7A is the $\overline{\text{THERM}}$ timer limit register. This 8-bit register allows a limit from 0 sec (first $\overline{\text{THERM}}$ assertion) to 5.825 sec to be set before an $\overline{\text{SMBALERT}}$ is generated. The $\overline{\text{THERM}}$ timer value is compared with the contents of the $\overline{\text{THERM}}$ timer limit register. If the $\overline{\text{THERM}}$ timer value exceeds the $\overline{\text{THERM}}$ timer limit value, the F4P bit (Bit 5) of Interrupt Status Register 2 is set and an $\overline{\text{SMBALERT}}$ is generated. The F4P bit (Bit 5) of Interrupt Mask Register 2 (0x75) masks out the $\overline{\text{SMBALERT}}$ if this bit is set to 1; however, the F4P bit of Interrupt Status Register 2 still is set if the $\overline{\text{THERM}}$ timer limit is exceeded.

Figure 33 is a functional block diagram of the $\overline{\text{THERM}}$ timer, limit, and associated circuitry. Writing a value of 0x00 to the $\overline{\text{THERM}}$ timer limit register (0x7A) causes an $\overline{\text{SMBALERT}}$ to be generated on the first $\overline{\text{THERM}}$ assertion. A $\overline{\text{THERM}}$ timer limit value of 0x01 generates an $\overline{\text{SMBALERT}}$ once cumulative $\overline{\text{THERM}}$ assertions exceed 45.52 ms.

### Configuring the $\overline{\text{THERM}}$ Behavior

1. Configure Pin 9 as a $\overline{\text{THERM}}$ timer input. Setting Bit 1 ($\overline{\text{THERM}}$ timer enable) of Configuration Register 3 (0x78) enables the $\overline{\text{THERM}}$ timer monitoring functionality. This is disabled on Pin 9 by default. Setting Bit 0 and Bit 1 (PIN9FUNC) of Configuration Register 4 (0x7D) enables $\overline{\text{THERM}}$ timer/output functionality on Pin 9 (Bit 1 of Configuration Register 3, $\overline{\text{THERM}}$, must also be set). Pin 9 can also be used as TACH4. Setting Bit 5, Bit 6, and Bit 7 of Configuration Register 5 (0x7C) makes $\overline{\text{THERM}}$ bidirectional. This means that if the appropriate temperature channel exceeds the $\overline{\text{THERM}}$ temperature limit, the $\overline{\text{THERM}}$ output asserts. If the ADT7473 is not pulling $\overline{\text{THERM}}$ low, but $\overline{\text{THERM}}$ is pulled low by an external device (such as a CPU overtemperature signal), the $\overline{\text{THERM}}$ timer also times $\overline{\text{THERM}}$ assertions. If Bit 5, Bit 6, and Bit 7 of Configuration Register 5 (0x7C) are set to 0, $\overline{\text{THERM}}$ is set as a timer input only.
2. Select the desired fan behavior for $\overline{\text{THERM}}$ timer events. Assuming the fans are running, setting Bit 2 (BOOST) of Configuration Register 3 (0x78) causes all fans to run at 100% duty cycle whenever $\overline{\text{THERM}}$ is asserted. This allows fail-safe system cooling. If this bit is 0, the fans run at their current settings and are not affected by $\overline{\text{THERM}}$ events. If the fans are not already running when $\overline{\text{THERM}}$ is asserted, the fans do not run at full speed.
3. Select whether $\overline{\text{THERM}}$ timer events should generate $\overline{\text{SMBALERT}}$ interrupts. Bit 5 (F4P) of Interrupt Mask Register 2 (0x75), when set, masks out the $\overline{\text{SMBALERT}}$ when the $\overline{\text{THERM}}$ timer limit value is exceeded. This bit should be cleared if $\overline{\text{SMBALERT}}$is based on $\overline{\text{THERM}}$ events required.

4. Select a suitable $\overline{\text{THERM}}$ limit value. This value determines whether an $\overline{\text{SMBALERT}}$ is generated on the first $\overline{\text{THERM}}$ assertion, or only if a cumulative $\overline{\text{THERM}}$ assertion time limit is exceeded. A value of 0x00 causes an $\overline{\text{SMBALERT}}$ to be generated on the first $\overline{\text{THERM}}$ assertion.
5. Select a $\overline{\text{THERM}}$ monitoring time. This value specifies how often OS or BIOS level software checks the $\overline{\text{THERM}}$ timer. For example, BIOS could read the $\overline{\text{THERM}}$ timer once an hour to determine the cumulative $\overline{\text{THERM}}$ assertion time. If, for example, the total $\overline{\text{THERM}}$ assertion time is <22.76 ms in Hour 1, >182.08 ms in Hour 2, and >5.825 sec in Hour 3, this can indicate that system performance is degrading significantly because $\overline{\text{THERM}}$ is asserting more frequently on an hourly basis.

Alternatively, OS- or BIOS-level software can timestamp when the system is powered on. If an $\overline{\text{SMBALERT}}$ is generated due to the $\overline{\text{THERM}}$ timer limit being exceeded, another timestamp can be taken. The difference in time can be calculated for a fixed $\overline{\text{THERM}}$ timer limit time. For example, if it takes one week for a $\overline{\text{THERM}}$ timer limit of 2.914 seconds to be exceeded and the next time it takes only one hour, this is an indication of a serious degradation in system performance.

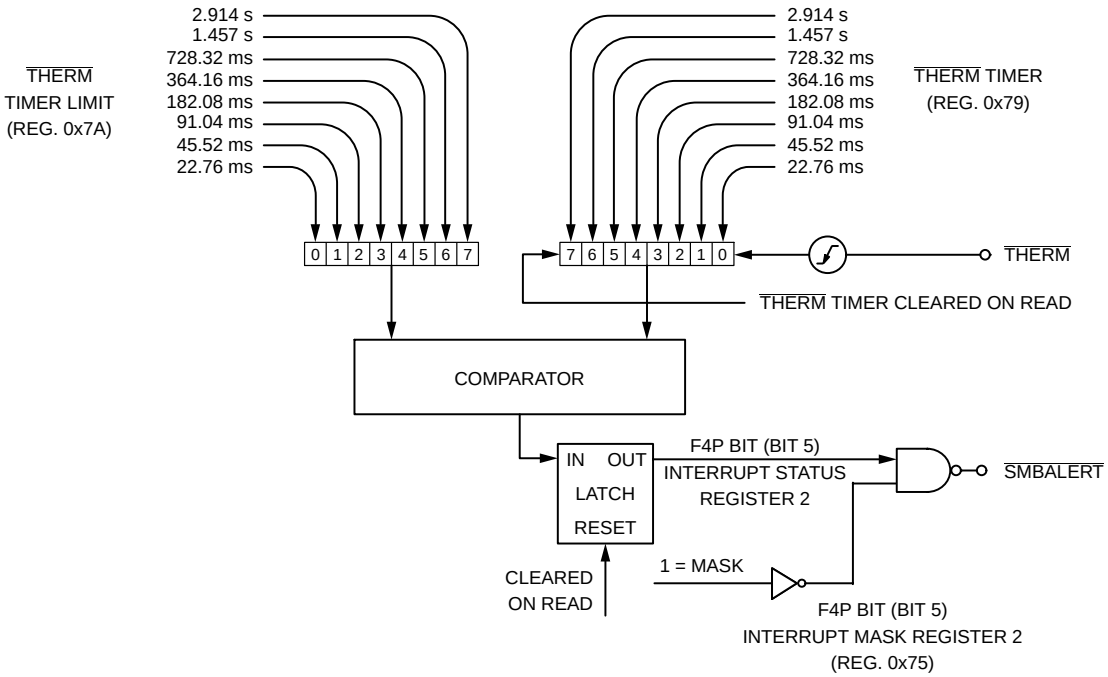



**Figure 33. Functional Diagram of the ADT7473 $\overline{\text{THERM}}$ Monitoring Circuitry**

### Configuring the $\overline{\text{THERM}}$ Pin as Bidirectional

In addition to monitoring $\overline{\text{THERM}}$ as an input, the ADT7473/ADT7473−1 can optionally drive $\overline{\text{THERM}}$ low as an output. When $\overline{\text{PROCHOT}}$ is bidirectional, $\overline{\text{THERM}}$ can be used to throttle the processor by asserting $\overline{\text{PROCHOT}}$. The user can preprogram system-critical thermal limits. If the temperature exceeds a thermal limit by 0.25°C, $\overline{\text{THERM}}$ asserts low. If the temperature is still above the thermal limit on the next monitoring cycle, $\overline{\text{THERM}}$ stays low. $\overline{\text{THERM}}$ remains asserted low until the temperature is equal to or below the thermal limit. Because the temperature for that channel is measured only once for every monitoring cycle after $\overline{\text{THERM}}$ asserts, it is guaranteed to remain low for at least one monitoring cycle.

The $\overline{\text{THERM}}$ pin can be configured to assert low, if the Remote 1, Local, or Remote 2 $\overline{\text{THERM}}$ temperature limits are exceeded by 0.25°C. The $\overline{\text{THERM}}$ temperature limit registers are at Register 0x6A, Register 0x6B, and Register 0x6C, respectively. Setting Bit 5, Bit 6, and Bit 7 of Configuration Register 5 (0x7C) makes $\overline{\text{THERM}}$ bidirectional for the Remote 1, Local, and Remote 2 temperature channels, respectively. Figure 34 shows how the $\overline{\text{THERM}}$ pin asserts low as an output in the event of a critical overtemperature.

An alternative method of disabling $\overline{\text{THERM}}$ is to program the $\overline{\text{THERM}}$ temperature limit to –64°C or less in Offset 64 mode, or −128°C or less in twos complement mode; that is, for $\overline{\text{THERM}}$ temperature limit values less than –63°C or –128°C, respectively, $\overline{\text{THERM}}$ is disabled. $\overline{\text{THERM}}$ can also be disabled by setting Bit 1 of Configuration Register 3 (0x78) to 0.

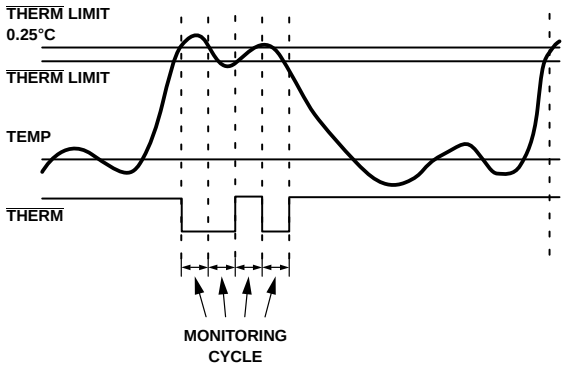



**Figure 34. Asserting $\overline{\text{THERM}}$ as an Output, Based on Tripping $\overline{\text{THERM}}$ Limits**

### Fan Drive Using PWM Control

The ADT7473/ADT7473−1 uses pulse-width modulation (PWM) to control fan speed. This relies on varying the duty cycle (or on/off ratio) of a square wave applied to the fan to vary the fan speed. The external circuitry required to drive a fan using PWM control is extremely simple. For 4-wire fans, the PWM drive might need only a pullup resistor. In many cases, the 4-wire fan PWM input has a built-in pullup resistor.

The ADT7473/ADT7473−1 PWM frequency can be set to a selection of low frequencies or a single high PWM frequency. The low frequency options are usually used for 3-wire fans, while the high frequency option is usually used with 4-wire fans.

Note that care must be taken to ensure that the PWM or TACH pins are not connected to a pullup supply greater than 3.6 V.

Many fans have internal pullups connected to the TACH/PWM pins to a supply greater than 3.6 V. Clamping or dividing down the voltage on these pins must be done where necessary. Clamping these pins with a Zener diode can also help prevent back-EMF related noise from being coupled into the system.

For 3-wire fans, a single N-channel MOSFET is the only drive device required. The specifications of the MOSFET depend on the maximum current required by the fan being driven. Typical notebook fans draw a nominal 170 mA; therefore, SOT devices can be used where board space is a concern. In desktops, fans can typically draw 250 mA to 300 mA each. If you drive several fans in parallel from a single PWM output or drive larger server fans, the MOSFET must handle the higher current requirements. The only other stipulation is that the MOSFET have a gate voltage drive, $V_{GS}$ < 3.3 V, for direct interfacing to the PWM output. The MOSFET should also have a low on resistance to ensure that there is not significant voltage drop across the FET, which would reduce the voltage applied across the fan and, therefore, the maximum operating speed of the fan.

Figure 35 shows how to drive a 3 wire fan using PWM control.

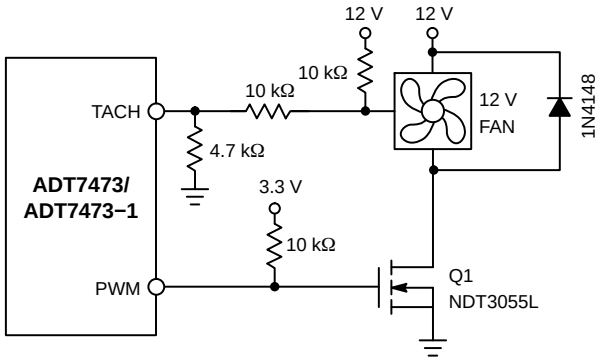



**Figure 35. Driving a 3-wire Fan Using an N-channel MOSFET**

Figure 35 uses a 10 kΩ pullup resistor for the TACH signal. This assumes that the TACH signal is an open-collector from the fan. In all cases, the TACH signal from the fan must be kept below 3.6 V maximum to prevent damaging the ADT7473/ADT7473−1. If uncertain as to whether the fan used has an open-collector or totem pole TACH output, use one of the input signal conditioning circuits shown in the Fan Speed Measurement section.

Figure 36 shows a fan drive circuit using an NPN transistor such as a general-purpose MMBT2222. While these devices are inexpensive, they tend to have much lower current handling capabilities and higher on resistance than MOSFETs. When choosing a transistor, care should be taken to ensure that it meets the fan's current requirements.

Ensure that the base resistor is chosen so that the transistor is saturated when the fan is powered on.

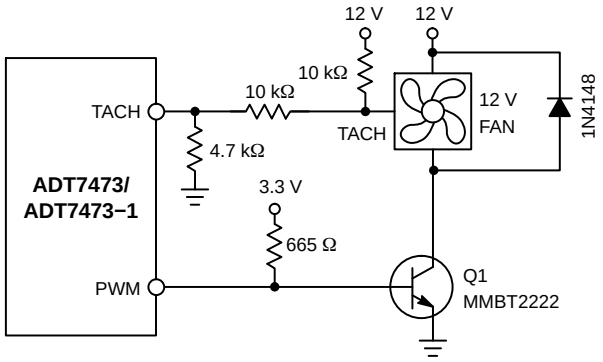



**Figure 36. Driving a 3-wire Fan Using an NPN Transistor**

Because 4-wire fans are powered continuously, the fan speed is not switched on or off as with previous PWM driven/powered fans. This enables it to perform better than 3-wire fans, especially for high frequency applications.

Figure 37 shows a typical drive circuit for 4-wire fans. As the PWM input on 4-wire fans is usually internally pulled up to a voltage greater than 3.6 V (the maximum voltage allowed on the ADT7473/ADT7473−1 PWM output), the PWM output should be clamped to 3.3 V using a Zener diode.

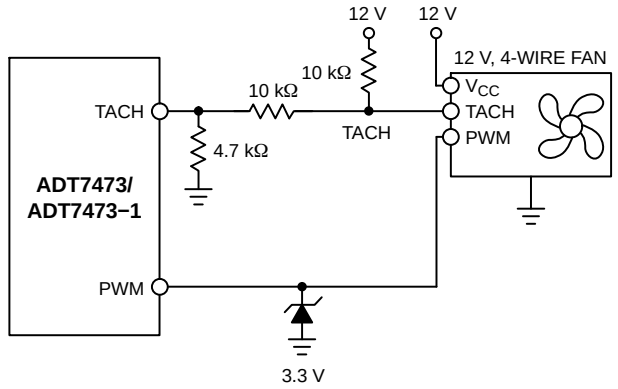



**Figure 37. Driving a 4-wire Fan**

### Driving Two Fans from PWM3

The ADT7473/ADT7473−1 has four TACH inputs available for fan speed measurement, but only three PWM drive outputs. If a fourth fan is used in the system, it should be driven from the PWM3 output in parallel with the third fan. Figure 38 shows how to drive two fans in parallel using low cost NPN transistors. Figure 39 shows the equivalent circuit using a MOSFET.

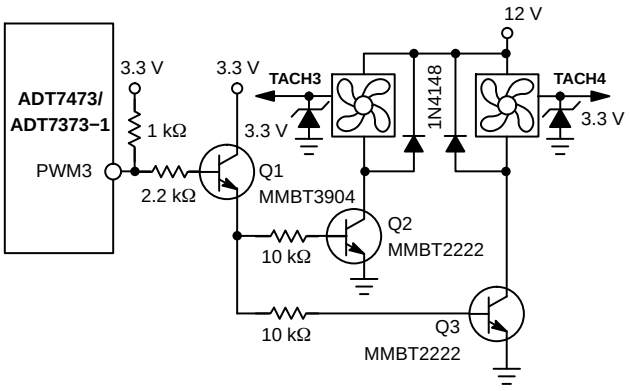



**Figure 38. Interfacing Two Fans in Parallel to the PWM3 Output Using Low Cost NPN Transistors**

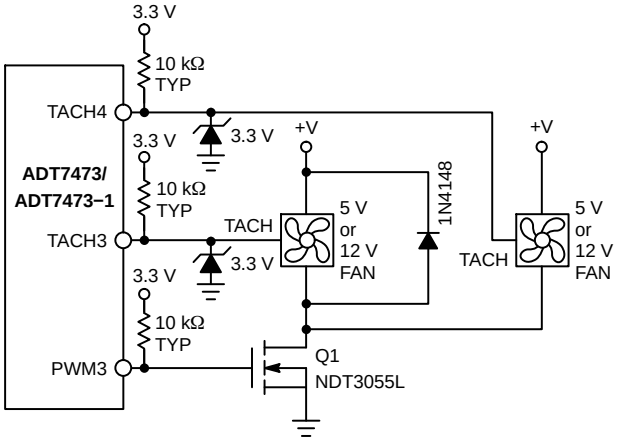



**Figure 39. Interfacing Two Fans in Parallel to the PWM3 Output Using a Single N-channel MOSFET**

Because the MOSFET can handle up to 3.5 A, it is simply a matter of connecting another fan directly in parallel with the first. Care should be taken in designing drive circuits with transistors and FETs to ensure the PWM pins are not required to source current and that they sink less than the 8 mA maximum current specified on the data sheet.

### Driving up to Three Fans from PWM3

TACH measurements for fans are synchronized to particular PWM channels; for example, TACH1 is synchronized to PWM1. TACH3 and TACH4 are both synchronized to PWM3, so PWM3 can drive two fans. Alternatively, PWM3 can be programmed to synchronize TACH2, TACH3, and TACH4 to the PWM3 output. This allows PWM3 to drive two or three fans. In this case, the drive circuitry looks the same, as shown in Figure 38 and Figure 39. The SYNC bit in Register 0x62 enables this function.

Synchronization is not required in high frequency mode when used with 4-wire fans.

**Table 33. SYNC: ENHANCED ACOUSTICS REGISTER 1 (REG. 0X62)**

| Bit | Mnemonic | Description |
|---|---|---|
| <4> | SYNC | 1 Synchronizes TACH2, TACH3, and TACH4 to PWM3. |

### TACH Inputs

Pin 4, Pin 6, Pin 7, and Pin 9 (when configured as TACH inputs) are open-drain TACH inputs intended for fan speed measurement.

Signal conditioning in the ADT7473/ADT7473−1 accommodates the slow rise and fall times typical of fan tachometer outputs. The maximum input signal range is 0 V to 3.6 V. In the event that these inputs are supplied from fan outputs that exceed 0 V to 3.6 V, either resistive attenuation of the fan signal or diode clamping must be included to keep inputs within an acceptable range.

Figure 40 to Figure 43 show circuits for most common fan TACH outputs.

If the fan TACH output has a resistive pullup to $V_{CC}$, it can be connected directly to the fan input, as shown in Figure 40.

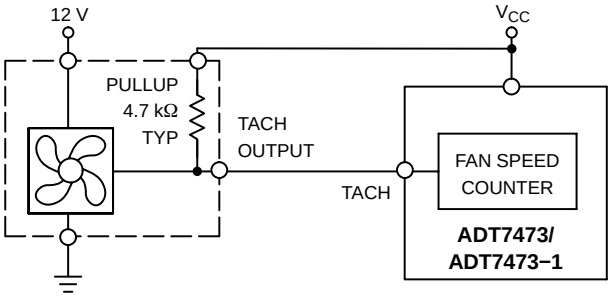



**Figure 40. Fan with TACH Pullup to $V_{CC}$**

If the fan output has a resistive pullup to 12 V (or other voltage greater than 3.6 V), the fan output can be clamped with a Zener diode, as shown in Figure 41. The Zener diode voltage should be chosen so that it is greater than $V_{IH}$ of the TACH input, but less than 3.6 V, allowing for the voltage tolerance of the Zener. A value of between 3.0 V and 3.6 V is suitable.

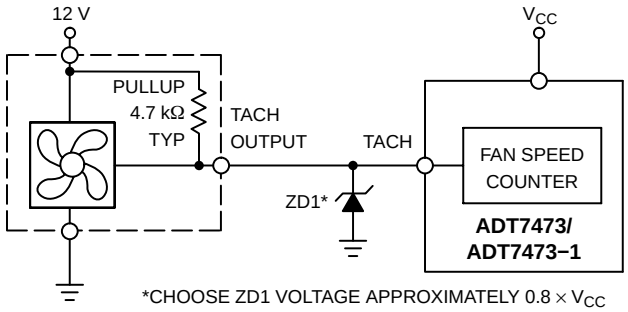



**Figure 41. Fan with TACH Pullup to Voltage > 3.6 V, Clamped with Zener Diode**

If the fan has a strong pullup (less than 1 kΩ) to 12 V or a totem-pole output, a series resistor can be added to limit the Zener current, as shown in Figure 42.

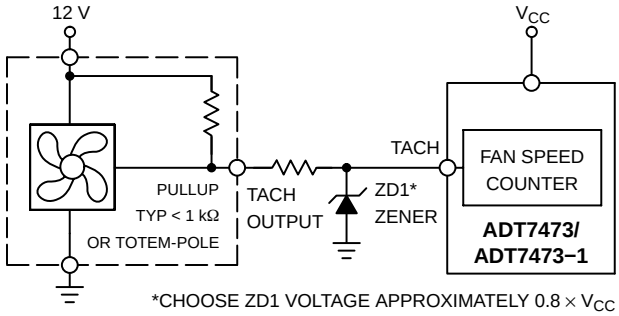



**Figure 42. Fan with Strong TACH. Pullup to > $V_{CC}$ or Totem-Pole Output, Clamped with Zener and Resistor**

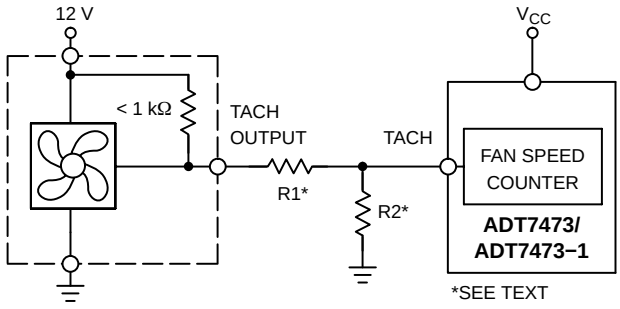



**Figure 43. Fan with Strong TACH. Pullup to > $V_{CC}$ or Totem-Pole Output, Attenuated with R1/R2**

Alternatively, a resistive attenuator can be used, as shown in Figure 43. R1 and R2 should be chosen such that

$$2\ V < V_{PULLUP} \times R2/(R_{PULLUP} + R1 + R2) < 3.6\ V \quad \text{(eq. 3)}$$

The fan inputs have an input resistance of nominally 160 kΩ to ground, which should be taken into account when calculating resistor values.

With a pullup voltage of 12 V and pullup resistor less than 1 kΩ, suitable values for R1 and R2 are 120 kΩ and 47 kΩ, respectively. This gives a high input voltage of 3.35 V.

### Fan Speed Measurement

The fan counter does not count the fan TACH output pulses directly, because the fan speed could be less than 1,000 RPM and it would take several seconds to accumulate a reasonably large and accurate count. Instead, the period of the fan revolution is measured by gating an on-chip 90 kHz oscillator into the input of a 16-bit counter for N periods of the fan TACH output (see Figure 44), so the accumulated count is actually proportional to the fan tachometer period, and inversely proportional to the fan speed.

N, the number of pulses counted, is determined by the settings of the TACH pulses per revolution register (Register 0x7B). This register contains two bits for each fan, allowing one, two (default), three, or four TACH pulses to be counted.

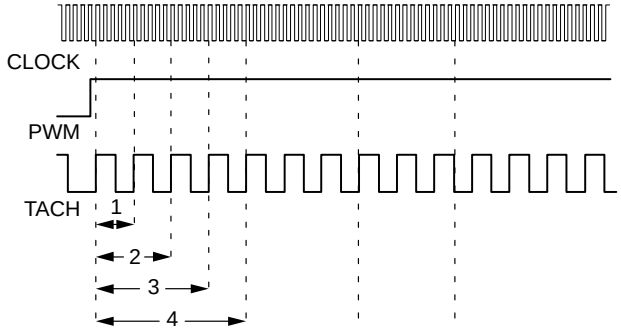



**Figure 44. Fan Speed Measurement**

### Fan Speed Measurement Registers

The fan tachometer readings are 16-bit values consisting of a 2-byte read from the ADT7473/ADT7473−1.

**Table 34. FAN SPEED MEASUREMENT REGISTERS**

| Register | Description | Default |
|---|---|---|
| 0x28 | TACH1 Low Byte | 0x00 |
| 0x29 | TACH1 High Byte | 0x00 |
| 0x2A | TACH2 Low Byte | 0x00 |
| 0x2B | TACH2 High Byte | 0x00 |
| 0x2C | TACH3 Low Byte | 0x00 |
| 0x2D | TACH3 High Byte | 0x00 |
| 0x2E | TACH4 Low Byte | 0x00 |
| 0x2F | TACH4 High Byte | 0x00 |

### Reading Fan Speed from the ADT7473/ADT7473−1

The measurement of fan speeds involves a 2-register read for each measurement. The low byte should be read first. This causes the high byte to be frozen until both high and low byte registers have been read, preventing erroneous TACH readings. The fan tachometer reading registers report back the number of 11.11 μs period clocks (90 kHz oscillator) gated to the fan speed counter, from the rising edge of the first fan TACH pulse to the rising edge of the third fan TACH

pulse (assuming two pulses per revolution are being counted). Because the device is essentially measuring the fan TACH period, the higher the count value, the slower the fan is actually running. A 16-bit fan tachometer reading of 0xFFFF indicates either the fan has stalled or is running very slowly (<100 RPM).

### High Limit > Comparison Performed

Because the actual fan TACH period is measured, falling below a fan TACH limit by 1 sets the appropriate status bit and can be used to generate an $\overline{\text{SMBALERT}}$.

### Measuring Fan TACH

When the ADT7473/ADT7473−1 starts up, TACH measurements are locked. In effect, an internal read of the low byte has been made for each TACH input. The net result of this is that all TACH readings are locked until the high byte is read from the corresponding TACH registers. All TACH related interrupts are also ignored until the appropriate high byte is read.

Once the corresponding high byte has been read, TACH measurements are unlocked and interrupts are processed as normal.

### Fan TACH Limit Registers

The fan TACH limit registers are 16-bit values consisting of two bytes.

**Table 35. FAN TACH LIMIT REGISTERS**

| Register | Description | Default |
|---|---|---|
| 0x54 | TACH1 Minimum Low Byte | 0xFF |
| 0x55 | TACH1 Minimum High Byte | 0xFF |
| 0x56 | TACH2 Minimum Low Byte | 0xFF |
| 0x57 | TACH2 Minimum High Byte | 0xFF |
| 0x58 | TACH3 Minimum Low Byte | 0xFF |
| 0x59 | TACH3 Minimum High Byte | 0xFF |
| 0x5A | TACH4 Minimum Low Byte | 0xFF |
| 0x5B | TACH4 Minimum High Byte | 0xFF |

### Fan Speed Measurement Rate

The fan TACH readings are normally updated once every second.

The FAST bit (Bit 3) of Configuration Register 3 (0x78), when set, updates the fan TACH readings every 250 ms.

If any of the fans are not being driven by a PWM channel but are powered directly from 5.0 V or 12 V, their associated dc bit in Configuration Register 3 should be set. This allows TACH readings to be taken on a continuous basis for fans connected directly to a dc source. For optimal results, the associated dc bit should always be set when using 4-wire fans.

### Calculating Fan Speed

Assuming a fan has two pulses per revolution (and two pulses per revolution being measured), fan speed is calculated by:

$$\text{Fan Speed (RPM)} = (90{,}000 \times 60)/\text{Fan TACH Reading}$$

where Fan TACH Reading is the 16-bit fan tachometer reading.

### Example

TACH1 High Byte (Register 0x29) = 0x17

TACH1 Low Byte (Register 0x28) = 0xFF

What is Fan 1 speed in RPM?

Fan 1 TACH Reading = 0x17FF = 6143 (decimal)

$\text{RPM} = (f \times 660)/\text{Fan 1 TACH Reading}$

$\text{RPM} = (90000 \times 660)/6143$

Fan Speed = 879 RPM

### Fan Pulses per Revolution

Different fan models can output either one, two, three, or four TACH pulses per revolution. Once the number of fan TACH pulses has been determined, it can be programmed into the fan pulses per revolution register (Register 0x7B) for each fan. Alternatively, this register can be used to determine the number or pulses per revolution output by a given fan. By plotting fan speed measurements at a 100% speed with different pulses per revolution setting, the smoothest graph with the lowest ripple determines the correct pulses per revolution value.

**Table 36. TACH PULSES/REVOLUTION REGISTER (REG. 0X7B)**

| Bit | Mnemonic | Description |
|---|---|---|
| <1:0> | FAN1 Default | 2 Pulses per Revolution |
| <3:2> | FAN2 Default | 2 Pulses per Revolution |
| <5:4> | FAN3 Default | 2 Pulses per Revolution |
| <7:6> | FAN4 Default | 2 Pulses per Revolution |

**Table 37. TACH PULSES/REVOLUTION REGISTER BIT VALUES**

| Value | Description |
|---|---|
| 00 | 1 Pulse per Revolution |
| 01 | 2 Pulses per Revolution |
| 10 | 3 Pulses per Revolution |
| 11 | 4 Pulses per Revolution |

### Fan Spin-up

The ADT7473/ADT7473−1 has a unique fan spin-up function. It spins the fan at 100% PWM duty cycle until two TACH pulses are detected on the TACH input. Once two TACH pulses are detected, the PWM duty cycle goes to the expected running value, for example, 33%. The advantage is that fans have different spin-up characteristics and take different times to overcome inertia. The ADT7473/ADT7473−1 runs the fans just fast enough to overcome inertia and is quieter on spin-up than fans programmed for a given spin-up time.

### Fan Startup Timeout

To prevent the generation of false interrupts as a fan spins up (because it is below running speed), the ADT7473/ ADT7473−1 includes a fan startup timeout function. During this time, the ADT7473/ADT7473−1 looks for two TACH pulses. If two TACH pulses are not detected, an interrupt is generated. Using Configuration Register 1 (0x40), Bit 5 (FSPDIS), this functionality can be changed (see the Disabling Fan Startup Timeout section).

**Table 38. PWM1 TO PWM3 CONFIGURATION (REG. 0X5C TO 0X5E)**

| Bit | Mnemonic | Description |
|---|---|---|
| <2:0> | SPIN | These bits control the startup time-out for PWM1 (0x5C), PWM2 (0x5D), PWM3 (0x5E).<br>000 = No Startup Timeout<br>001 = 100 ms<br>010 = 250 ms (Default)<br>011 = 400 ms<br>100 = 667 ms<br>101 = 1 s<br>110 = 2 s<br>111 = 4 s |

### Disabling Fan Startup Timeout

Although fan startup makes fan spin-ups much quieter than fixed-time spin-ups, the option exists to use fixed spin-up times. Setting Bit 5 (FSPDIS) to 1 in Configuration Register 1 (0x40) disables the spin-up for two TACH pulses. Instead, the fan spins up for the fixed time as selected in Register 0x5C to Register 0x5E.

### PWM Logic State

The PWM outputs can be programmed high for a 100% duty cycle (non-inverted) or low for a 100% duty cycle (inverted).

**Table 39. PWM1 TO PWM3 CONFIGURATION (REG. 0X5C TO 0X5E) BITS**

| Bit | Mnemonic | Description |
|---|---|---|
| 4 | INV | 0 = logic high for 100% PWM duty cycle<br>1 = logic low for 100% PWM duty cycle |

### Low Frequency Mode PWM Drive Frequency

The PWM drive frequency can be adjusted for the application. Register 0x5F to Register 0x61 configure the PWM frequency for PWM1 to PWM3, respectively. In high frequency mode, the PWM drive frequency is always 22.5 kHz.

### High Frequency Mode PWM Drive

Setting Bit 3 of Register 0x5F, 60H or 61H enables high frequency mode for fans 1, 2 and 3.

**Table 40. PWM FREQUENCY REGISTERS (REG. 0X5F TO 0X61)**

| Bit | Mnemonic | Description |
|---|---|---|
| <2:0> | FREQ | 000 = 11.0 Hz<br>001 = 14.7 Hz<br>010 = 22.1 Hz<br>011 = 29.4 Hz<br>100 = 35.3 Hz (Default)<br>101 = 44.1 Hz<br>110 = 58.8 Hz<br>111 = 88.2 Hz |

### Fan Speed Control

The ADT7473/ADT7473−1 controls fan speed using automatic and manual modes.

In automatic fan speed control mode, fan speed is automatically varied with temperature and without CPU intervention, once initial parameters are set up. The advantage of this is that, if the system hangs, the user is guaranteed the system is protected from overheating. The automatic fan speed control incorporates a feature called dynamic $T_{MIN}$ calibration. This feature reduces the design effort required to program the automatic fan speed control loop. For more information and procedures on how to program the automatic fan speed control loop and dynamic $T_{MIN}$ calibration, see the Programming the Automatic Fan Speed Control Loop section.

In manual fan speed control mode, the ADT7473/ ADT7473−1 allows the duty cycle of any PWM output to be manually adjusted. This can be useful if the user wants to change fan speed in software or adjust the PWM duty cycle output for test purposes. Bits <7:5> of Register 0x5C to Register 0x5E (PWM configuration registers) control the behavior of each PWM output.

**Table 41. PWM CONFIGURATION (REG. 0X5C TO 0X5E) BITS**

| Bit | Mnemonic | Description |
|---|---|---|
| <7:5> | BHVR 111 | Manual Mode |

Once under manual control, each PWM output can be manually updated by writing to Register 0x30 to Register 0x32 (PWM current duty cycle registers).

### Programming the PWM Current Duty Cycle Registers

The PWM current duty cycle registers are 8-bit registers that allow the PWM duty cycle for each output to be set anywhere from 0% to 100% in steps of 0.39%.

The value to be programmed into the $PWM_{MIN}$ register is given by:

$$\text{Value (decimal)} = PWM_{MIN}/0.39 \quad \text{(eq. 4)}$$

### Example 1

For a PWM duty cycle of 50%

Value (decimal) = 50/0.39 = 128 (decimal)

Value = 128 (decimal) or 0x80 (hex)

### Example 2

For a PWM duty cycle of 33%

Value (decimal) = 33/0.39 = 85 (decimal)

Value = 85 (decimal) or 0x54 (hex)

**Table 42. PWM CURRENT DUTY CYCLE REGISTERS**

| Register | Description | Default |
|---|---|---|
| 0x30 | PWM1 Duty Cycle | 0xFF (0%) |
| 0x31 | PWM2 Duty Cycle | 0xFF (0%) |
| 0x32 | PWM3 Duty Cycle | 0xFF (0%) |

By reading the PWMx current duty cycle registers, the user can keep track of the current duty cycle on each PWM output, even when the fans are running in automatic fan speed control mode or acoustic enhancement mode. See the Programming the Automatic Fan Speed Control Loop section for details.

### Fan Presence Detect

This feature can be used to determine if a 4-wire fan is directly connected to a PWM output. This feature does not work for 3-wire fans. To detect whether a 4-wire fan is connected directly to a PWM output, the following steps must be performed in this order:

1. Drive the appropriate PWM outputs to 100% duty cycle.
2. Set Bit 0 of Configuration Register 2 (0x73).
3. Wait 5 ms.
4. Program the fans to run at a different speed if necessary.
5. Read the state of Bits <3:1> of Configuration Register 2 (0x73). The state of these bits reflects whether a 4-wire fan is directly connected to the PWM output.

As the detection time only takes 5 ms, programming the PWM outputs to 100% and then back to their normal speed is not noticeable in most cases.

### Description of How Fan Presence Detect Works

Typical 4-wire fans have an internal pull up to 4.75 V ±10%, which typically sources 5 mA. While the detection cycle is on, an internal current sink is turned on, sinking current from the fan's internal pullup. By driving some of the current from the fan's internal pullup (~100 μA), the logic buffer switches to a defined logic state. If this state is high, a fan is present; if it is low, no fan is present.

The PWM input voltage should be clamped to 3.3 V. This ensures the PWM output is not pulled to a voltage higher than the maximum allowable voltage on that pin (3.6 V).

### Sleep States

The ADT7473/ADT7473−1 has been specifically designed to operate from a 3.3 V STBY supply. In computers that support S3 and S5 states, the core voltage of the processor is lowered in these states. If using the dynamic $T_{MIN}$ mode, lowering the core voltage of the processor changes the CPU temperature and the dynamics of the system under dynamic $T_{MIN}$ control. Likewise, when monitoring $\overline{\text{THERM}}$, the $\overline{\text{THERM}}$ timer should be disabled during these states.

### Dynamic $T_{MIN}$ Control Register 1 (0X36)

### Bit <1> $V_{CCP}$LO = 1

When the $V_{CCP}$ voltage drops below the $V_{CCP}$ low limit, the following occurs:

1. Status Bit 1 ($V_{CCP}$) in Status Register 1 is set.
2. $\overline{\text{SMBALERT}}$ is generated, if enabled.
3. $\overline{\text{THERM}}$ monitoring is disabled. The $\overline{\text{THERM}}$ timer should hold its value prior to the S3 or S5 state.
4. Dynamic $T_{MIN}$ control is disabled. This prevents $T_{MIN}$ from being adjusted due to an S3 or S5 state.
5. The ADT7473/ADT7473−1 is prevented from entering the shutdown state.

Once the core voltage, $V_{CCP}$, goes above the $V_{CCP}$ low limit, everything is re-enabled, and the system resumes normal operation.

### XNOR Tree Test Mode

The ADT7473/ADT7473−1 includes an XNOR tree test mode. This mode is useful for in-circuit test equipment at board-level testing. By applying stimulus to the pins included in the XNOR tree, it is possible to detect opens or shorts on the system board.

Figure 45 shows the signals that are exercised in the XNOR tree test mode. The XNOR tree test is invoked by setting Bit 0 (XEN) of the XNOR tree test enable register (0x6F).

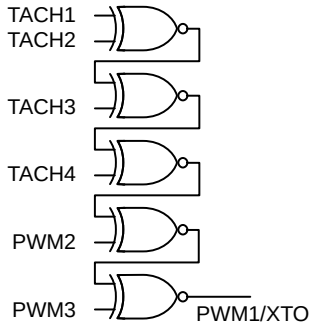



**Figure 45. XNOR Tree Test**

### Power-on Default

When the ADT7473 is powered up, it polls the $V_{CCP}$ input. By default, the ADT7473−1 powers up with fans running, eliminating the need for polling of $V_{CCP}$.

If $V_{CCP}$ stays below 0.75 V (the system CPU power rail is not powered up), the ADT7473 assumes the functionality of the default registers after the ADT74731 is addressed via any valid SMBus transaction.

If $V_{CC}$ goes high (the system processor power rail is powered up), a fail-safe timer begins to count down. If the ADT7473 is not addressed by any valid SMBus transactions before the fail-safe timeout (4.6 seconds) lapses, the ADT7473 drives the fans to full speed. If the ADT7473 is addressed by a valid SMBus transaction after this point, the fans stop, and the ADT7473 assumes its default settings and begins normal operation.

If $V_{CCP}$ goes high (the system processor power rail is powered up), then a fail-safe timer begins to count down. If the ADT7473 is addressed by a valid SMBus transaction before the fail-safe timeout (4.6 seconds) lapses, then the ADT7473 operates normally, assuming the functionality of all the default registers. See the flow chart in Figure 46.

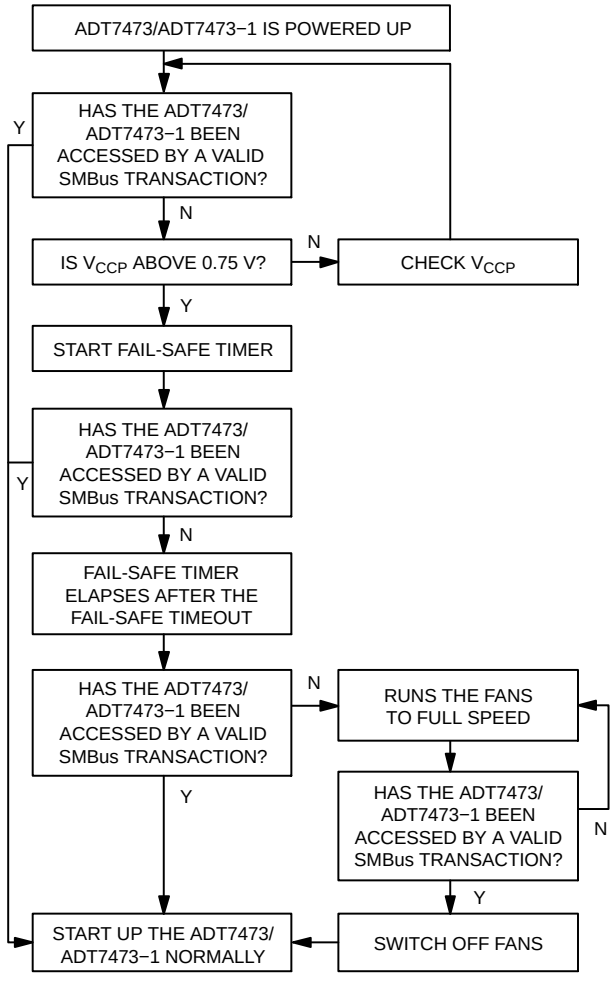



**Figure 46. Power-on Flow Chart**

### Programming the Automatic Fan Speed Control Loop

To understand the automatic fan speed control loop, it is strongly recommended to use the ADT7473/ADT7473−1 evaluation board and software while reading this section.

This section provides the system designer with an understanding of the automatic fan control loop, and provides step-by-step guidance on effectively evaluating and selecting critical system parameters. To optimize the system characteristics, the designer needs to consider the system configuration, including the number of fans, where they are located, and what temperatures are measured in the particular system.

The mechanical or thermal engineer who is tasked with the system thermal characterization should also be involved at the beginning of the process.

### Automatic Fan Control Overview

The ADT7473/ADT7473−1 can automatically control the speed of fans based on the measured temperature. This is done independently of CPU intervention once initial parameters are set up.

The ADT7473/ADT7473−1 has a local temperature sensor and two remote temperature channels that can be connected to a CPU on-chip thermal diode (available on Intel Pentium class CPUs and other CPUs). These three temperature channels can be used as the basis for automatic fan speed control to drive fans using PWM.

Automatic fan speed control reduces acoustic noise by optimizing fan speed according to accurately measured temperature. Reducing fan speed can also decrease system current consumption. The automatic fan speed control mode is very flexible due to the number of programmable parameters, including $T_{MIN}$ and $T_{RANGE}$. The $T_{MIN}$ and $T_{RANGE}$ values for a temperature channel and, therefore, for a given fan, are critical because they define the thermal characteristics of the system. The thermal validation of the system is one of the most important steps in the design process, so these values should be selected carefully.

Figure 47 gives a top-level overview of the automatic fan control circuitry on the ADT7473/ADT7473−1. From a systems-level perspective, up to three system temperatures can be monitored and used to control three PWM outputs. The three PWM outputs can be used to control up to four fans. The ADT7473/ADT7473−1 allows the speed of four fans to be monitored. Each temperature channel has a thermal calibration block, allowing the designer to individually configure the thermal characteristics of each temperature channel. For example, a designer can decide to run the CPU fan when CPU temperature increases above 60°C, and a chassis fan when the local temperature increases above 45°C. At this stage, the designer has not assigned these thermal calibration settings to a particular fan drive (PWM) channel. The right side of Figure 47 shows controls that are fan-specific. The designer has individual control over parameters such as minimum PWM duty cycle, fan

speed failure thresholds, and even ramp control of the PWM outputs. Automatic fan control, then, ultimately allows graceful fan speed changes that are less perceptible to the system user.

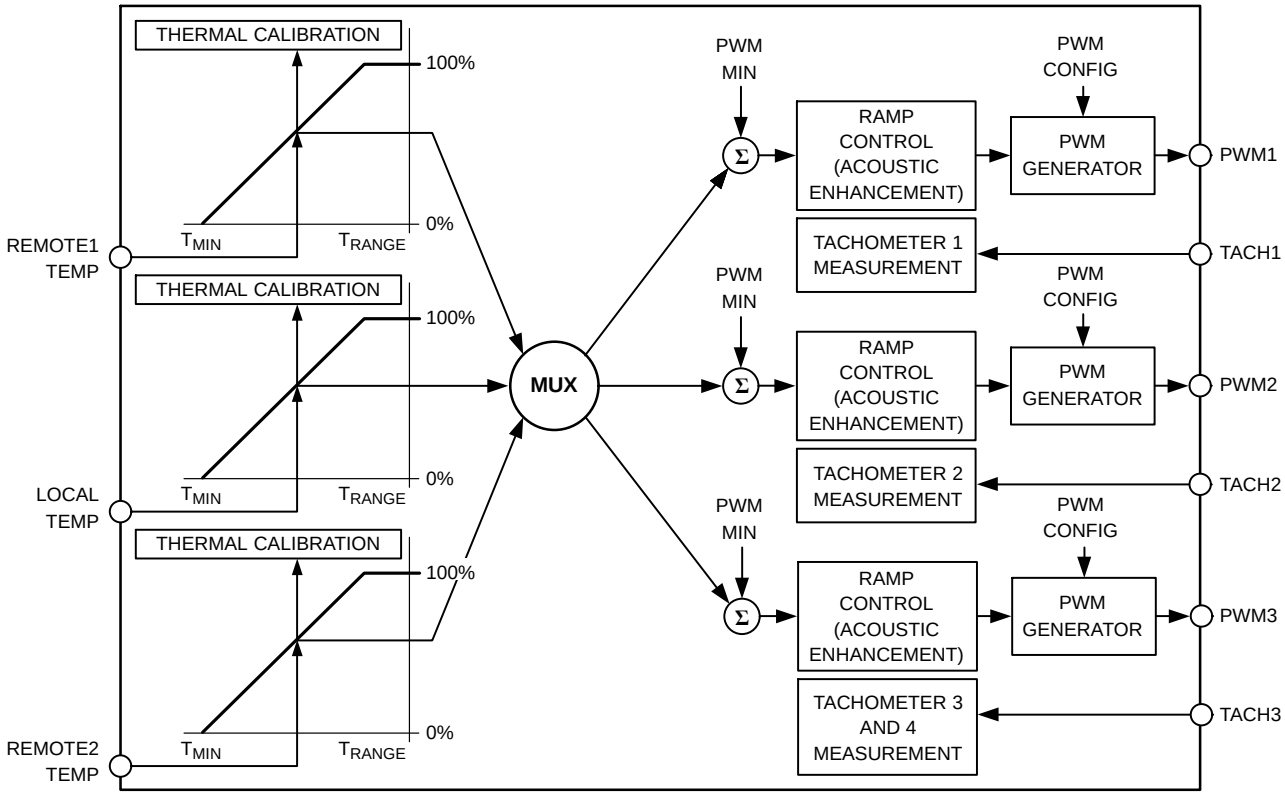



**Figure 47. Automatic Fan Control Block Diagram**

**Step 1: Hardware Configuration**

During system design, the motherboard sensing and control capabilities should be addressed early in the design stages. Decisions about how these capabilities are used should involve the system thermal/mechanical engineer. Consider the following questions:

1. What ADT7473/ADT7473−1 functionality will be used?
   - PWM2 or $\overline{\text{SMBALERT}}$ for ADT7473?
   - $\overline{\text{THERM\_LATCH}}$ or PWM2 for ADT7473−1?
   - TACH4 fan speed measurement or over-temperature $\overline{\text{THERM}}$ function?

   The ADT7473/ ADT7473−1 offers multifunctional pins that can be reconfigured to suit different system requirements and physical layouts. These multifunction pins are software programmable.
2. How many fans will be supported in the system, three or four?

   This influences the choice of whether to use the TACH4 pin or to reconfigure it for the $\overline{\text{THERM}}$ function.
3. Is the CPU fan to be controlled using the ADT7473/ADT7473−1 or will it run at full speed 100% of the time?

   If run at full speed, 100% of the time, this frees up a PWM output, but the system is louder.
4. Where will the ADT7473/ADT7473−1 be physically located in the system?

   This influences the assignment of the temperature measurement channels to particular system thermal zones. For example, locating the ADT7473/ADT7473−1 close to the VRM controller circuitry allows the VRM temperature to be monitored using the local temperature channel.

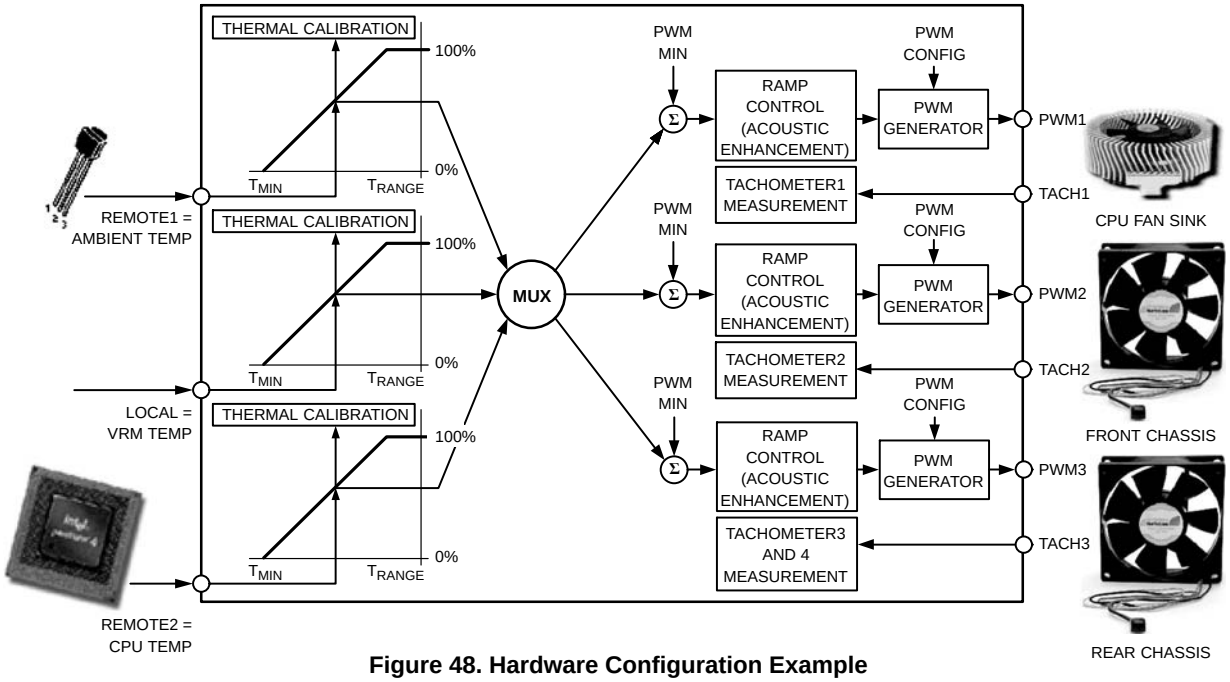



**Figure 48. Hardware Configuration Example**

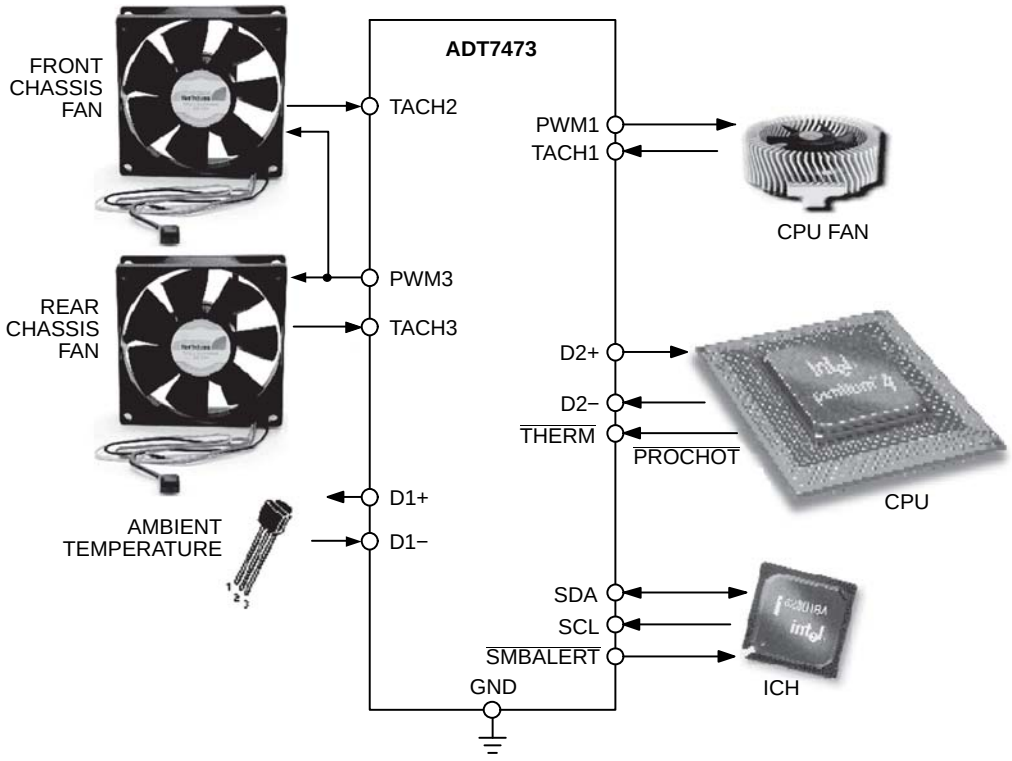



**Figure 49. Recommended Implementation 1**

### Recommended Implementation 1

Configuring the ADT7473, as in Figure 49 provides the system designer with the following features:

- Two PWM outputs for fan control of up to three fans. (The front and rear chassis fans are connected in parallel.)
- Three TACH fan speed measurement inputs.
- $V_{CC}$ measured internally through Pin 4.
- CPU core voltage measurement ($V_{CORE}$).
- VRM temperature using local temperature sensor.
- CPU temperature measured using the Remote 1 temperature channel.
- Ambient temperature measured through the Remote 2 temperature channel.
- Bidirectional $\overline{\text{THERM}}$ pin allows the monitoring of $\overline{\text{PROCHOT}}$ output from an Intel® Pentium® 4 processor, for example, or can be used as an overtemperature $\overline{\text{THERM}}$ output.
- $\overline{\text{SMBALERT}}$ system interrupt output.

### Step 2: Configuring the Mux

After the system hardware configuration is determined, the fans can be assigned to particular temperature channels. Not only can fans be assigned to individual channels, but the behavior of the fans is also configurable. For example, fans can be run under automatic fan control, manually (under software control), or at the fastest speed calculated by multiple temperature channels. The mux is the bridge between temperature measurement channels and the three PWM outputs.

Bits <7:5> (BHVR) of Register 0x5C, Register 0x5D, and Register 0x5E (PWM configuration registers) control the behavior of the fans connected to the PWM1, PWM2, and PWM3 outputs. The values selected for these bits determine how the mux connects a temperature measurement channel to a PWM output.

### Automatic Fan Control Mux Options

Bits <7:5> (BHVR), Register 0x5C, Register 0x5D, Register 0x5E.

000 = Remote 1 temperature controls PWMx
001 = Local temperature controls PWMx
010 = Remote 2 temperature controls PWMx
101 = Fastest speed calculated by local and Remote 2 temperature controls PWMx
110 = Fastest speed calculated by all three temperature channel controls PWMx

The fastest speed calculated options pertain to controlling one PWM output based on multiple temperature channels. The thermal characteristics of the three temperature zones can be set to drive a single fan. An example is the fan turning on when Remote 1 temperature exceeds 60°C, or if the local temperature exceeds 45°C.

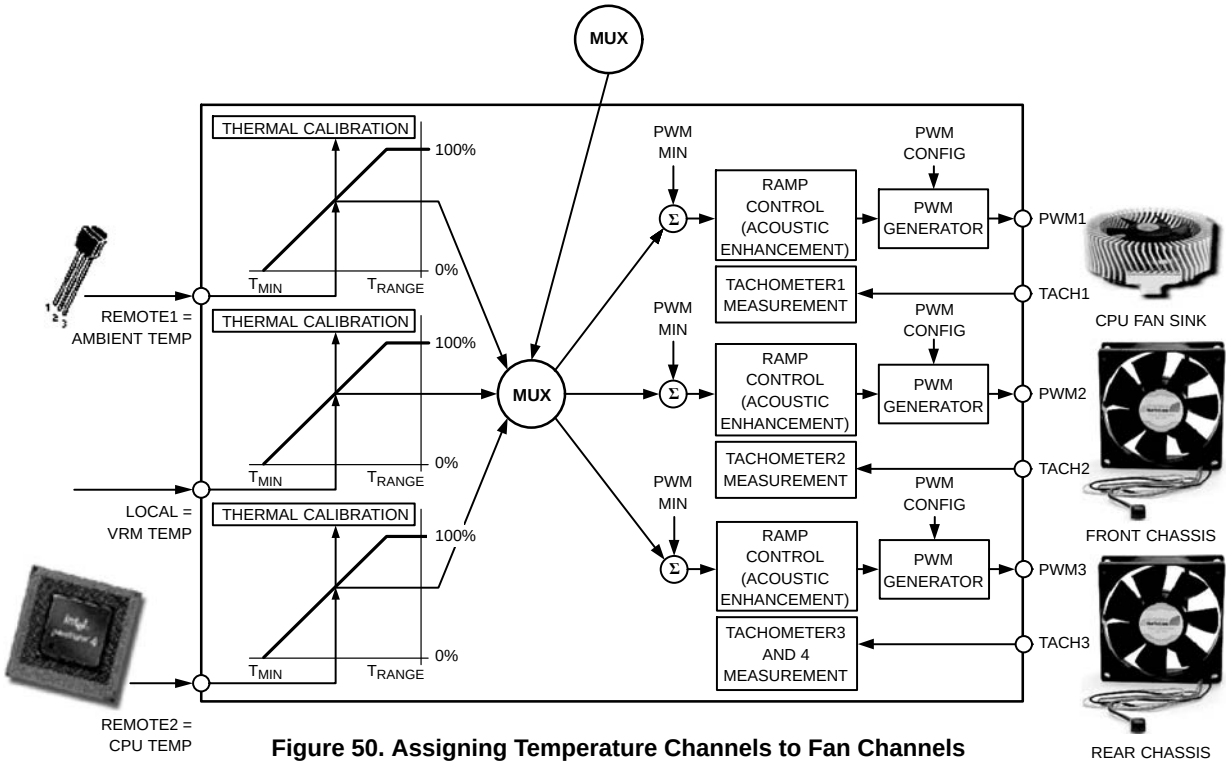



**Figure 50. Assigning Temperature Channels to Fan Channels**

### Other Mux Options

Bits <7:5> (BHVR), Register 0x5C, Register 0x5D, Register 0x5E.

011 = PWMx runs full speed (default for ADT7473−1)
100 = PWMx disabled (default for ADT7473)
111 = manual mode

In normal mode, PWMx runs under software control. In this mode, PWM duty cycle registers (Register 0x30 to Register 0x32) are writable and control the PWM outputs.

### Mux Configuration Example

This is an example of how to configure the mux in a system using the ADT7473/ADT7473−1 to control three fans. The CPU fan sink is controlled by PWM1, the front chassis fan is controlled by PWM2, and the rear chassis fan is controlled by PWM3. The mux is configured for the following fan control behaviors:

- PWM1 (CPU fan sink) is controlled by the fastest speed calculated by the local (VRM temperature) and Remote 2 (processor) temperature. In this case, the CPU fan sink is also used to cool the VRM.
- PWM2 (front chassis fan) is controlled by the Remote 1 temperature (ambient).
- PWM3 (rear chassis fan) is controlled by the Remote 1 temperature (ambient).

### Example Mux Settings

Bits <7:5> (BHVR), PWM1 Configuration Register (0x5C)
101 = Fastest speed calculated by local and Remote 2 temperature controls PWM1

Bits <7:5> (BHVR), PWM2 Configuration Register (0x5D)
000 = Remote 1 temperature controls PWM2

Bits <7:5> (BHVR), PWM3 Configuration Register (0x5E)
000 = Remote 1 temperature controls PWM3

These settings configure the mux, as shown in Figure 51.

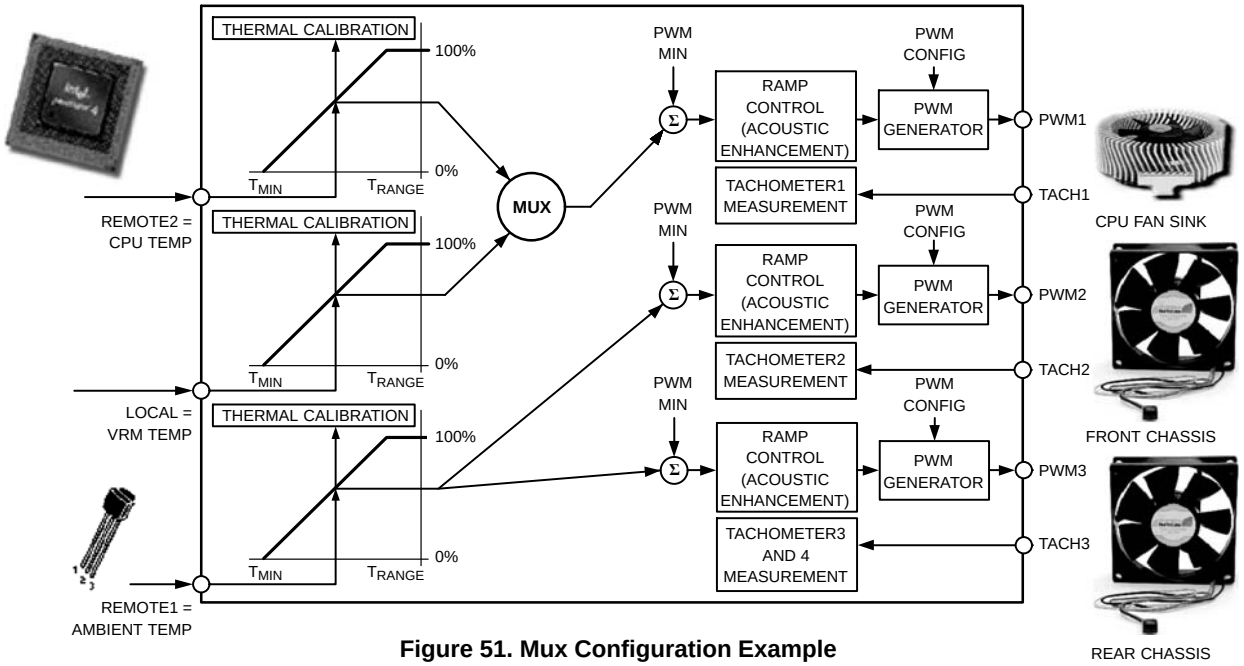



**Figure 51. Mux Configuration Example**

### Step 3: $T_{MIN}$ Settings for Thermal Calibration Channels

$T_{MIN}$ is the temperature at which the fans start to turn on under automatic fan control. The speed at which the fan runs at $T_{MIN}$ is programmed later. The $T_{MIN}$ values chosen are temperature channel specific, for example, 25°C for ambient channel, 30°C for VRM temperature, and 40°C for processor temperature.

$T_{MIN}$ is an 8-bit value, either twos complement or Offset 64, that can be programmed in 1°C increments. A $T_{MIN}$ register is associated with each temperature measurement channel: Remote 1 local and Remote 2 temperature. Once the $T_{MIN}$ value is exceeded, the fan turns on and runs at the minimum PWM duty cycle. The fan turns off once the temperature drops below $T_{MIN} - T_{HYST}$.

To overcome fan inertia, the fan is spun up until two valid TACH rising edges are counted. See the Fan Startup Timeout section for more details. In some cases, primarily for psycho-acoustic reasons, it is desirable that the fan never switches off below $T_{MIN}$. Bits <7:5> of Enhanced Acoustics Register 1 (0x62), when set, can keep the fans running at the PWM minimum duty cycle, if the temperature falls below $T_{MIN}$.

**Table 43. $T_{MIN}$ REGISTERS**

| Register | Description | Default |
|---|---|---|
| 0x67 | Remote 1 Temperature $T_{MIN}$ | 0x9A (90°C) |
| 0x68 | Local Temperature $T_{MIN}$ | 0x9A (90°C) |
| 0x69 | Remote 2 Temperature $T_{MIN}$ | 0x9A (90°C) |

**Enhanced Acoustics Register 1 (0x62)**

Bit 7 (MIN3) = 0, PWM3 is off (0% PWM duty cycle) when temperature is below $T_{MIN} - T_{HYST}$.

Bit 7 (MIN3) = 1, PWM3 runs at PWM3 minimum duty cycle below $T_{MIN} - T_{HYST}$.

Bit 6 (MIN2) = 0, PWM2 is off (0% PWM duty cycle) when temperature is below $T_{MIN} - T_{HYST}$.

Bit 6 (MIN2) = 1, PWM2 runs at PWM2 minimum duty cycle below $T_{MIN} - T_{HYST}$.

Bit 5 (MIN1) = 0, PWM1 is off (0% PWM duty cycle) when temperature is below $T_{MIN} - T_{HYST}$.

Bit 5 (MIN1) = 1, PWM1 runs at PWM1 minimum duty cycle below $T_{MIN} - T_{HYST}$.

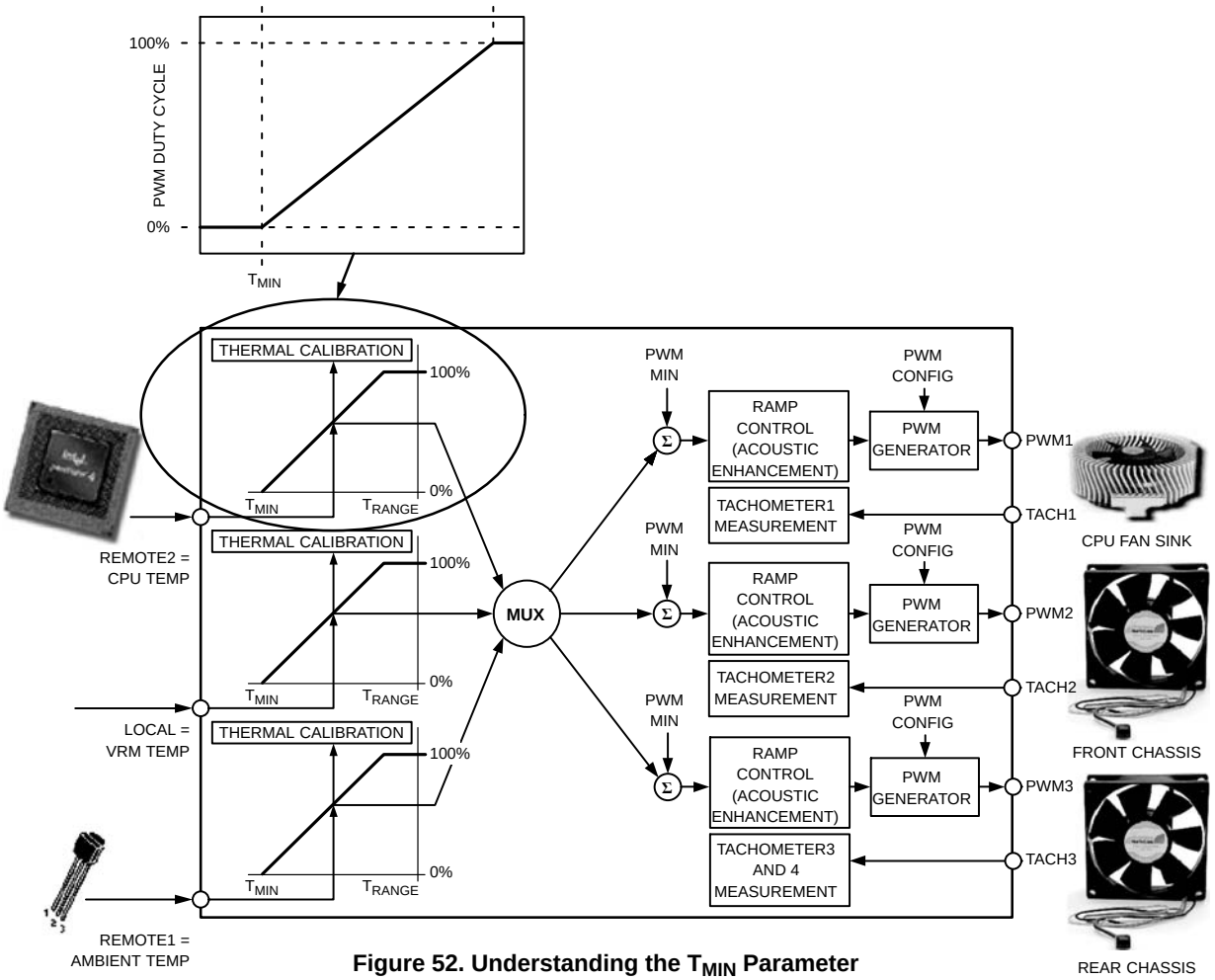



**Figure 52. Understanding the $T_{MIN}$ Parameter**

**Step 4: $PWM_{MIN}$ for Each PWM (Fan) Output**

$PWM_{MIN}$ is the minimum PWM duty cycle at which each fan in the system runs. It is also the start speed for each fan under automatic fan control once the temperature rises above $T_{MIN}$ (see Figure 53). For maximum system acoustic benefit, $PWM_{MIN}$ should be set as low as possible. Depending on the fan used, the $PWM_{MIN}$ setting is usually in the 20% to 33% duty cycle range. This value can be found through fan validation.

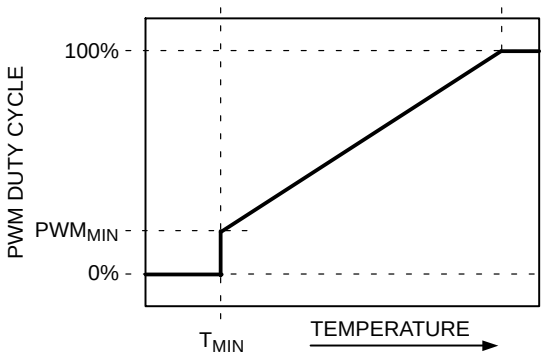



**Figure 53. $PWM_{MIN}$ Determines Minimum PWM Duty Cycle**

More than one PWM output can be controlled from a single temperature measurement channel. For example, Remote 1 temperature can control PWM1 and PWM2 outputs. If two different fans are used on PWM1 and PWM2, the fan characteristics can be set up differently. As a result, Fan 1 driven by PWM1 can have a different $PWM_{MIN}$ value than that of Fan 2 connected to PWM2. Figure 54 illustrates this as $PWM1_{MIN}$ (front fan) is turned on at a minimum duty cycle of 20%, while $PWM2_{MIN}$ (rear fan) turns on at a minimum of 40% duty cycle. However, both fans turn on at exactly the same temperature, defined by $T_{MIN}$.

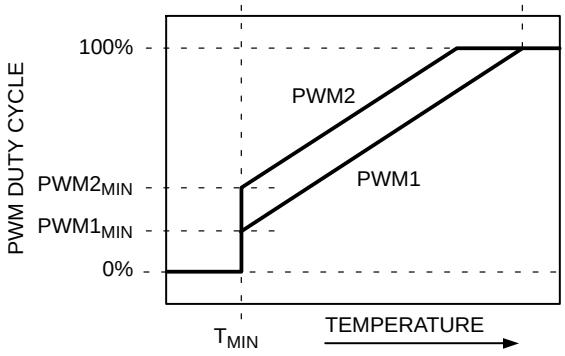



**Figure 54. Operating Two Different Fans from a Single Temperature Channel**

### Programming the $PWM_{MIN}$ Registers

The $PWM_{MIN}$ registers are 8-bit registers that allow the minimum PWM duty cycle for each output to be configured anywhere from 0% to 100%. This allows the minimum PWM duty cycle to be set in steps of 0.39%.

The value to be programmed into the $PWM_{MIN}$ register is given by:

Value (decimal) = $PWM_{MIN}/0.39$

### Example 1

For a minimum PWM duty cycle of 50%

Value (decimal) = 50/0.39 = 128 (decimal)

Value = 128 (decimal) or 80 (hex)

### Example 2

For a minimum PWM duty cycle of 33%

Value (decimal) = 33/0.39 = 85 (decimal)

Value = 85 (decimal) or 54 (hex)

**Table 44. $PWM_{MIN}$ REGISTERS**

| Register | Description | Default |
|---|---|---|
| 0x64 | PWM1 Minimum Duty Cycle | 0x80 (50%) |
| 0x65 | PWM2 Minimum Duty Cycle | 0x80 (50%) |
| 0x66 | PWM3 Minimum Duty Cycle | 0x80 (50%) |

### Note on Fan Speed and PWM Duty Cycle

The PWM duty cycle does not directly correlate to fan speed in RPM. Running a fan at 33% PWM duty cycle does not equate to running the fan at 33% speed. Driving a fan at 33% PWM duty cycle actually runs the fan at closer to 50% of its full speed. This is because fan speed in %RPM generally relates to the square root of PWM duty cycle. Given a PWM square wave as the drive signal, fan speed in RPM approximates to:

$$\%\ \text{fanspeed} = \sqrt{\text{PWM duty cycle} \times 10} \qquad \text{(eq. 5)}$$

### Step 5: $PWM_{MAX}$ for PWM (Fan) Outputs

$PWM_{MAX}$ is the maximum duty cycle at which each fan in the system runs under the automatic fan speed control loop. For maximum system acoustic benefit, $PWM_{MAX}$ should be as low as possible, but should be capable of maintaining the processor temperature limit at an acceptable level. If the $\overline{\text{THERM}}$ temperature limit is exceeded, the fans are still boosted to 100% for fail-safe cooling (see Figure 55).

There is a $PWM_{MAX}$ limit for each fan channel. The default value of this register is 0xFF and thus has no effect unless it is programmed.

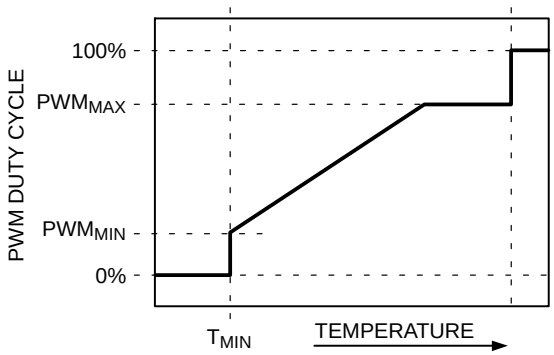



**Figure 55. $PWM_{MAX}$ Determines Maximum PWM Duty Cycle Below the $\overline{\text{THERM}}$ Temperature Limit**

### Programming the $PWM_{MAX}$ Registers

The $PWM_{MAX}$ registers are 8-bit registers that allow the maximum PWM duty cycle for each output to be configured anywhere from 0% to 100%. This allows the maximum PWM duty cycle to be set in steps of 0.39%.

The value to be programmed into the $PWM_{MAX}$ register is given by:

Value (decimal) = $PWM_{MAX}/0.39$

### Example 1

For a maximum PWM duty cycle of 50%

Value (decimal) = 50/0.39 = 128 (decimal)

Value = 128 (decimal) or 80 (hex)

### Example 2

For a minimum PWM duty cycle of 75%

Value (decimal) = 75/0.39 = 85 (decimal)

Value = 192 (decimal) or C0 (hex)

**Table 45. $PWM_{MAX}$ REGISTERS**

| Register | Description | Default |
|---|---|---|
| 0x38 | PWM1 Maximum Duty Cycle | 0xFF (100%) |
| 0x39 | PWM2 Maximum Duty Cycle | 0xFF (100%) |
| 0x3A | PWM3 Maximum Duty Cycle | 0xFF (100%) |

See the Note on Fan Speed and PWM Duty Cycle section.

### Step 6: $T_{RANGE}$ for Temperature Channels

$T_{RANGE}$ is the range of temperature over which automatic fan control occurs once the programmed $T_{MIN}$ temperature is exceeded. $T_{RANGE}$ is a temperature slope, not an arbitrary value, that is, a $T_{RANGE}$ of 40°C holds true only for $PWM_{MIN}$ = 33%. If $PWM_{MIN}$ is increased or decreased, the effective $T_{RANGE}$ changes. Refer to Figure 56.

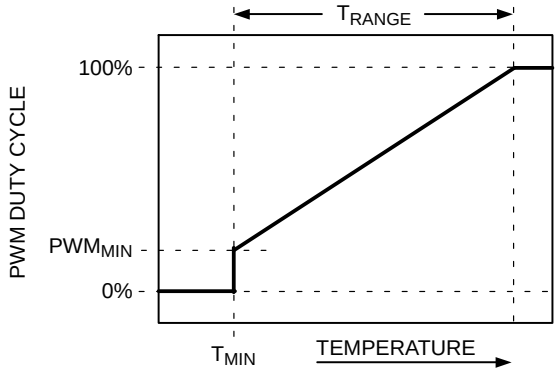



**Figure 56. $T_{RANGE}$ Parameter Affects Cooling Slope**

The $T_{RANGE}$ or fan control slope is determined by the following procedure:

1. Determine the maximum operating temperature for that channel (for example, 70°C).
2. Determine experimentally the fan speed (PWM duty cycle value) that does not exceed the temperature at the worst-case operating points (for example, 70°C is reached when the fans are running at 50% PWM duty cycle).
3. Determine the slope of the required control loop to meet these requirements.
4. Graphically program and visualize this functionality using the ADT7473/ADT7473−1 evaluation software.

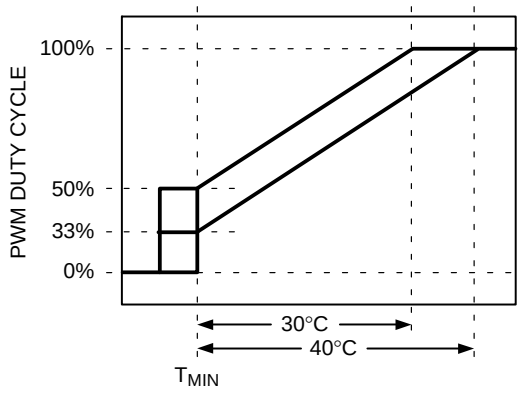



**Figure 57. Adjusting $PWM_{MIN}$ Affects $T_{RANGE}$**

Figure 57 shows how adjusting $PWM_{MIN}$ affects $T_{RANGE}$.

$T_{RANGE}$ is implemented as a slope, which means that as $PWM_{MIN}$ is changed, $T_{RANGE}$ changes, but the actual slope remains the same. The higher the $PWM_{MIN}$ value, the smaller the effective $T_{RANGE}$, that is, the fan reaches full speed (100%) at a lower temperature. Figure 58 shows how increasing $PWM_{MIN}$ changes the effective $T_{RANGE}$.

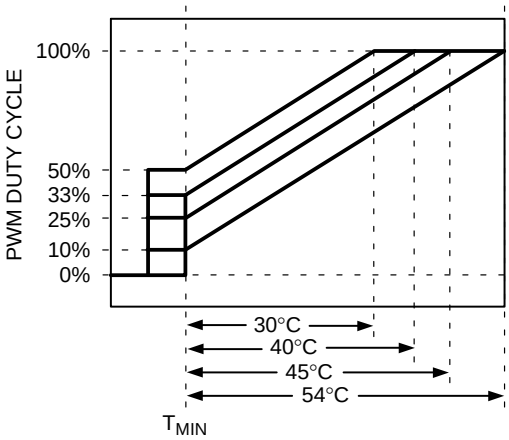



**Figure 58. Increasing $PWM_{MIN}$ Changes Effective $T_{RANGE}$**

For a given $T_{RANGE}$ value, the temperature at which the fan runs at full speed for different $PWM_{MIN}$ values can be easily calculated as follows:

$$T_{MAX} = T_{MIN} + (\text{Max DC} - \text{Min DC}) \times T_{RANGE} / 170$$

where:

$T_{MAX}$ is the temperature at which the fan runs full speed.
$T_{MIN}$ is the temperature at which the fan turns on.
Max DC is the maximum duty cycle (100%) = 255 decimal.
Min DC is equal to $PWM_{MIN}$.
$T_{RANGE}$ is the duty PWM duty cycle vs. temperature slope.

### Example 1

Calculate T, given that $T_{MIN}$ = 30°C, $T_{RANGE}$ = 40°C, and $PWM_{MIN}$ = 10% duty cycle = 26 (decimal).

$T_{MAX} = T_{MIN} + (\text{Max DC} - \text{Min DC}) \times T_{RANGE} / 170$
$T_{MAX}$ = 30°C + (100% − 10%) × 40°C/170
$T_{MAX}$ = 30°C + (255 − 26) × 40°C/170
$T_{MAX}$ = 84°C (effective $T_{RANGE}$ = 54°C)

### Example 2

Calculate $T_{MAX}$, given that $T_{MIN}$ = 30°C, $T_{RANGE}$ = 40°C, and $PWM_{MIN}$ = 25% duty cycle = 64 (decimal).

$T_{MAX} = T_{MIN} + (\text{Max DC} - \text{Min DC}) \times T_{RANGE} / 170$
$T_{MAX}$ = 30°C + (100% − 25%) × 40°C/170
$T_{MAX}$ = 30°C + (255 − 64) × 40°C/170
$T_{MAX}$ = 75°C (effective $T_{RANGE}$ = 45°C)

### Example 3

Calculate $T_{MAX}$, given that $T_{MIN}$ = 30°C, $T_{RANGE}$ = 40°C, and $PWM_{MIN}$ = 33% duty cycle = 85 (decimal).

$T_{MAX} = T_{MIN} + (\text{Max DC} - \text{Min DC}) \times T_{RANGE}/170$
$T_{MAX} = 30°C + (100\% - 33\%) \times 40°C/170$
$T_{MAX} = 30°C + (255 - 85) \times 40°C/170$
$T_{MAX} = 70°C$ (effective $T_{RANGE} = 40°C$)

### Example 4

Calculate $T_{MAX}$, given that $T_{MIN}$ = 30°C, $T_{RANGE}$ = 40°C, and $PWM_{MIN}$ = 50% duty cycle = 128 (decimal).

$T_{MAX} = T_{MIN} + (\text{Max DC} - \text{Min DC}) \times T_{RANGE}/170$
$T_{MAX} = 30°C + (100\% - 50\%) \times 40°C/170$
$T_{MAX} = 30°C + (255 - 128) \times 40°C/170$
$T_{MAX} = 60°C$ (effective $T_{RANGE} = 30°C$)

### Selecting a $T_{RANGE}$ Slope

The $T_{RANGE}$ value can be selected for each temperature channel: Remote 1, local, and Remote 2. Bits <7:4> ($T_{RANGE}$) of Register 0x5F to Register 0x61 define the $T_{RANGE}$ value for each temperature channel.

**Table 46. SELECTING A $T_{RANGE}$ VALUE**

| **Bits <7:4>** (Note 1) | **$T_{RANGE}$ (°C)** |
|---|---|
| 0000 | 2 |
| 0001 | 2.5 |
| 0010 | 3.33 |
| 0011 | 4 |
| 0100 | 5 |
| 0101 | 6.67 |
| 0110 | 8 |
| 0111 | 10 |
| 1000 | 13.33 |
| 1001 | 16 |
| 1010 | 20 |
| 1011 | 26.67 |
| 1100 | 32 (Default) |
| 1101 | 40 |
| 1110 | 53.33 |
| 1111 | 80 |

1. Register 0x5F configures Remote 1 $T_{RANGE}$; Register 0x60 configures local $T_{RANGE}$; Register 0x61 configures Remote 2 $T_{RANGE}$.

### Summary of $T_{RANGE}$ Function

When using the automatic fan control function, the temperature at which the fan reaches full speed can be calculated by:

$$T_{MAX} = T_{MIN} + T_{TRANGE} \quad \text{(eq. 6)}$$

Equation 6 holds true only when $PWM_{MIN}$ is equal to 33% PWM duty cycle.

Increasing or decreasing $PWM_{MIN}$ changes the effective $T_{RANGE}$, although the fan control still follows the same PWM duty cycle to temperature slope. The effective $T_{RANGE}$ for different $PWM_{MIN}$ values can be calculated using Equation 7.

$$T_{MAX} = T_{MIN} + (\text{Max DC} - \text{Min DC}) \times T_{TRANGE}/170 \quad \text{(eq. 7)}$$

where (Max DC − Min DC) × $T_{RANGE}$/170 is the effective $T_{RANGE}$ value.

See the Note on Fan Speed and PWM Duty Cycle section. Figure 59 shows PWM duty cycle vs. temperature for each $T_{RANGE}$ setting. The lower graph shows how each $T_{RANGE}$ setting affects fan speed vs. temperature. As indicated by the graph, the effect on fan speed is nonlinear.

The graphs in Figure 59 assume the fan starts from 0% PWM duty cycle. Clearly, the minimum PWM duty cycle, $PWM_{MIN}$, needs to be factored in to see how the loop actually performs in the system. Figure 60 shows how $T_{RANGE}$ is affected when the $PWM_{MIN}$ value is set to 20%. It can be seen that the fan actually runs at about 45% fan speed when the temperature exceeds $T_{MIN}$.

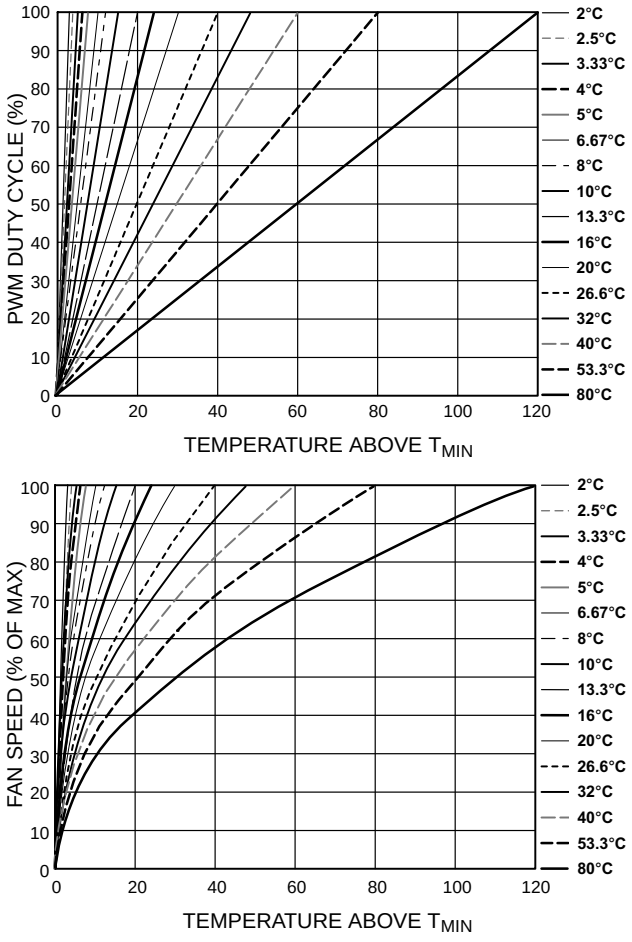



**Figure 59. $T_{RANGE}$ vs. Actual Fan Speed Profile**

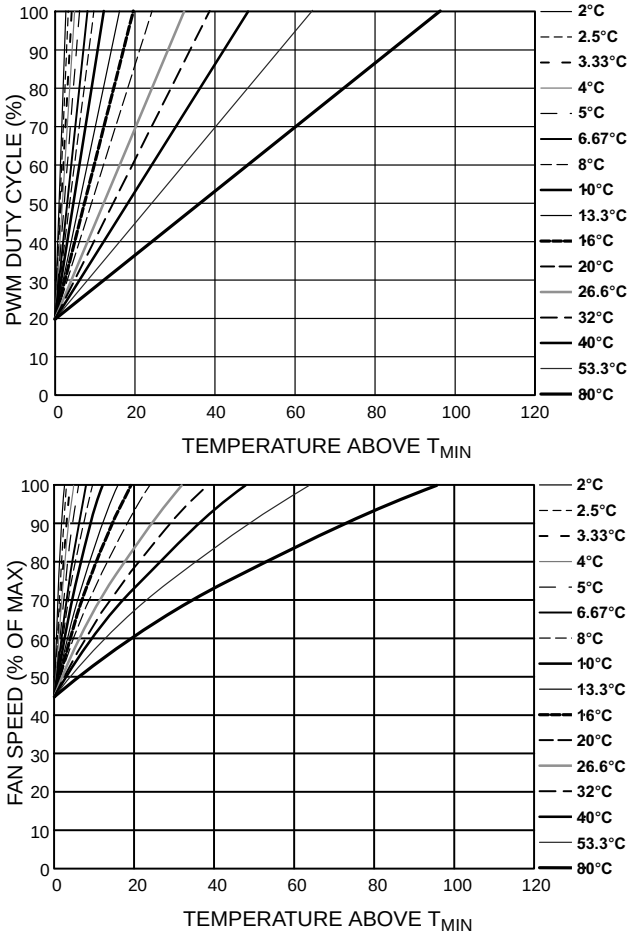



**Figure 60. $T_{RANGE}$ and % Fan Speed Slopes with $PWM_{MIN}$ = 20%**

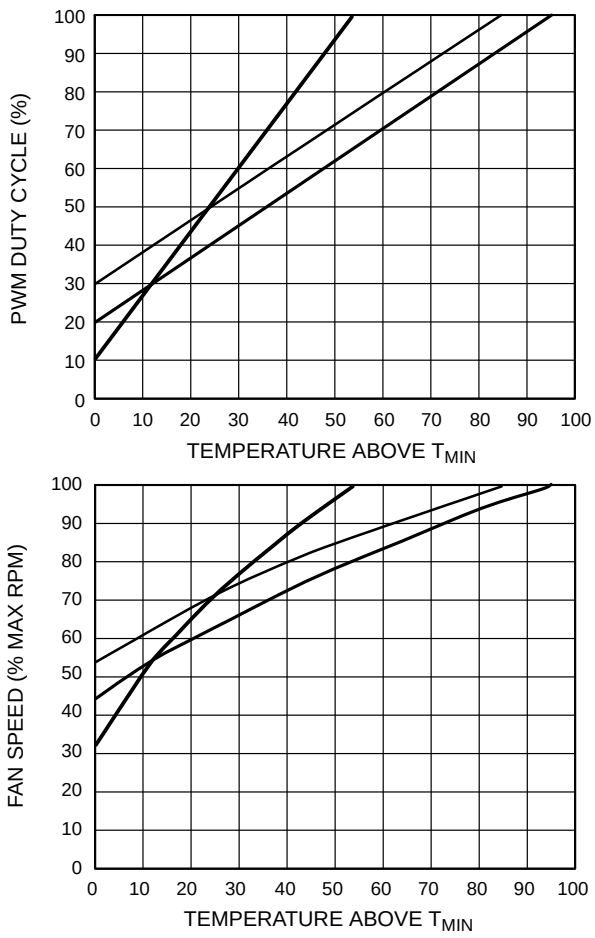



**Figure 61. $T_{RANGE}$ and % Fan Speed Slopes for VRM, Ambient, and CPU Temperature Channels**

### Example: Determining $T_{RANGE}$ for Each Temperature Channel

The following example shows how the different $T_{MIN}$ and $T_{RANGE}$ settings can be applied to three different thermal zones. In this example, the following $T_{RANGE}$ values apply:

$T_{RANGE}$ = 80°C for ambient temperature
$T_{RANGE}$ = 53.3°C for CPU temperature
$T_{RANGE}$ = 40°C for VRM temperature

This example uses the mux configuration described in the Step 2: Configuring the Mux section, with the ADT7473/ ADT7473−1 connected as shown in Figure 61. Both CPU temperature and VRM temperature drive the CPU fan connected to PWM1.

Ambient temperature drives the front chassis fan and rear chassis fan connected to PWM2 and PWM3. The front chassis fan is configured to run at $PWM_{MIN}$ = 20%. The rear chassis fan is configured to run at $PWM_{MIN}$ = 30%. The CPU fan is configured to run at $PWM_{MIN}$ = 10%.

### Note on 4-wire Fans

The control range for 4-wire fans is much wider than that for 3-wire fans. In many cases, 4-wire fans can start with a PWM drive of as little as 20%.

### Step 7: $T_{\overline{THERM}}$ for Temperature Channels

$T_{\overline{THERM}}$ is the absolute maximum temperature allowed on a temperature channel. When operating above this temperature, a component such as the CPU or VRM might be beyond its safe operating limit. When the temperature measured exceeds $T_{\overline{THERM}}$ all fans are driven at 100% PWM duty cycle (full speed) to provide critical system cooling.

The fans remain running at 100% until the temperature drops below $T_{\overline{THERM}}$ − hysteresis, where hysteresis is the number programmed into the hysteresis registers (Register 0x6D and Register 0x6E). The default hysteresis value is 4°C.

The $T_{\overline{THERM}}$ limit should be considered the maximum worst-case operating temperature of the system. Because exceeding any $T_{\overline{THERM}}$ limit runs all fans at 100%, it has very negative acoustic effects. Ultimately, this limit should be set up as a fail-safe, and it should not be exceeded under normal system operating conditions.

Note that the $T_{\overline{THERM}}$ limits are nonmaskable and affect the fan speed no matter how the automatic fan control settings are configured. This allows some flexibility because a $T_{RANGE}$ value can be selected based on its slope, while a

hard limit (such as 70°C), can be programmed as $T_{MAX}$ (the temperature at which the fan reaches full speed) by setting $T_{\overline{THERM}}$ to that limit (for example, 70°C).

**Table 47. $\overline{THERM}$ REGISTERS**

| Register | Description | Default |
|---|---|---|
| 0x6A | Remote 1 $\overline{THERM}$ Temperature Limit | 0xA4 (100°C) |
| 0x6B | Local $\overline{THERM}$ Temperature Limit | 0xA4 (100°C) |
| 0x6C | Remote 2 $\overline{THERM}$ Temperature Limit | 0xA4 (100°C) |

### Hysteresis Registers

Register 0x6D, Remote 1 Local Temperature Hysteresis Register

Bits <7:4> Remote 1 temperature hysteresis (4°C default)

Bits <3:0> Local temperature hysteresis (4°C default)

Register 0x6E, Remote 2 Temperature Hysteresis Register

Bits <7:4> Remote 2 temperature hysteresis (4°C default)

Because each hysteresis setting is four bits, hysteresis values are programmable from 1°C to 15°C. It is not recommended that hysteresis values be programmed to 0°C, because this disables hysteresis. In effect, this would cause the fans to cycle between normal speed and 100% speed, creating unsettling acoustic noise.

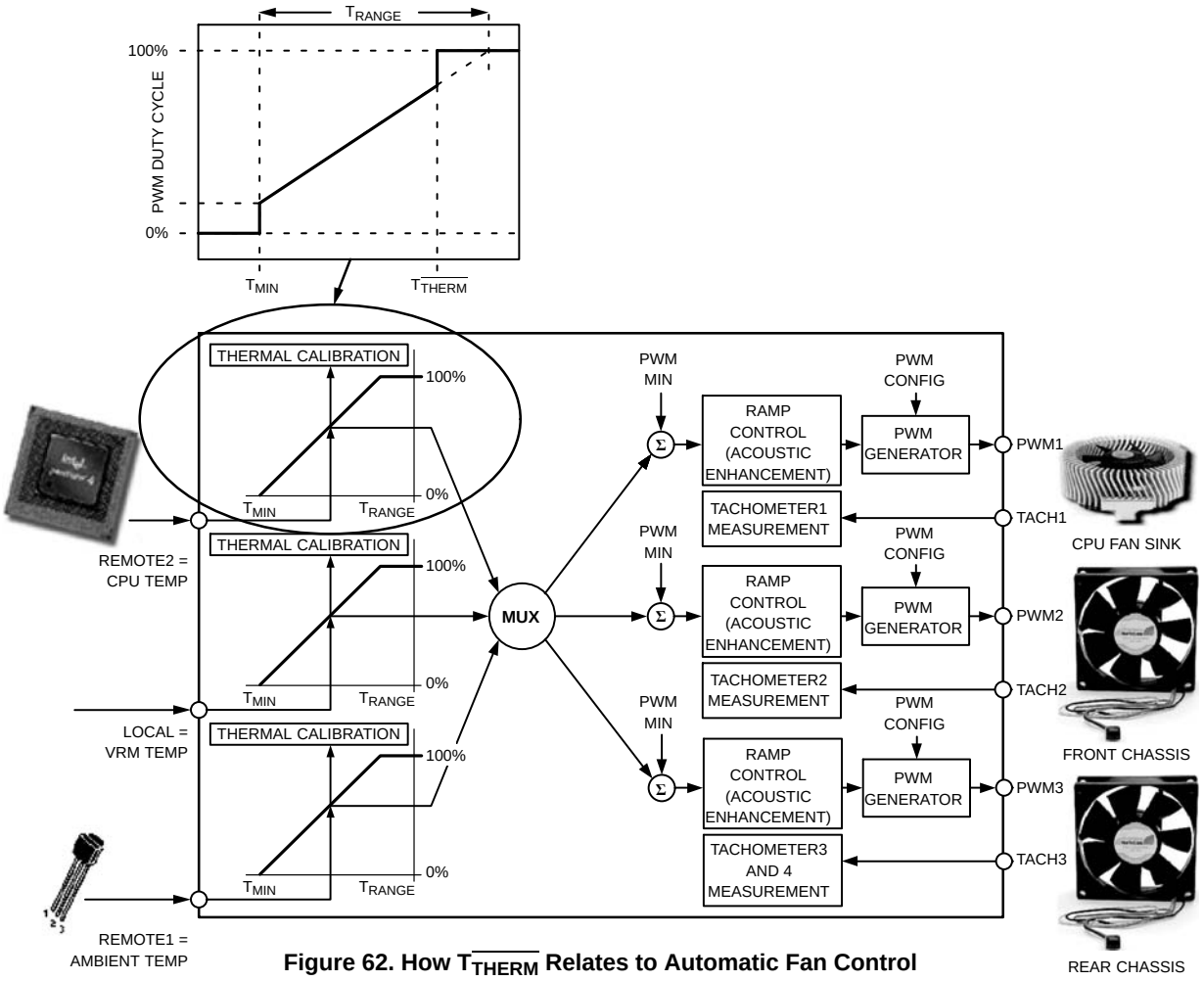



**Figure 62. How $T_{\overline{THERM}}$ Relates to Automatic Fan Control**

### Step 8: $T_{HYST}$ for Temperature Channels

$T_{HYST}$ is the amount of extra cooling a fan provides after the temperature measured has dropped back below $T_{MIN}$ before the fan turns off. The premise for temperature hysteresis ($T_{HYST}$) is that, without it, the fan would merely chatter or cycle on and off regularly whenever temperature is hovering at about the $T_{MIN}$ setting.

The $T_{HYST}$ value chosen determines the amount of time needed for the system to cool down or heat up as the fan turns on and off. Values of hysteresis are programmable in the range 1°C to 15°C. Larger values of $T_{HYST}$ prevent the fans from chattering on and off. The $T_{HYST}$ default value is set at 4°C.

The $T_{HYST}$ setting applies not only to the temperature hysteresis for fan on/off, but the same setting is used for the $T_{\overline{THERM}}$ hysteresis value, described in Step 6: $T_{RANGE}$ for Temperature Channels section. Therefore, programming Register 0x6D and Register 0x6E sets the hysteresis for both fan on/off and the $\overline{THERM}$ function.

### Hysteresis Registers

Register 0x6D, Remote 1, Local Hysteresis Register

Bits <7:4>, Remote 1 temperature hysteresis (4°C default)

Bits <3:0>, Local temperature hysteresis (4°C default)

Register 0x6E, Remote 2 Temperature Hysteresis Register

Bits <7:4>, Remote 2 temperature hysteresis (4°C default)

In some applications, it is required that fans not turn off below $T_{MIN}$, but remain running at $PWM_{MIN}$. Bits <7:5> of the Enhanced Acoustics Register 1 (0x62) allow the fans to be turned off or to be kept spinning below $T_{MIN}$. If the fans are always on, the $T_{HYST}$ value has no effect on the fan when the temperature drops below $T_{MIN}$.

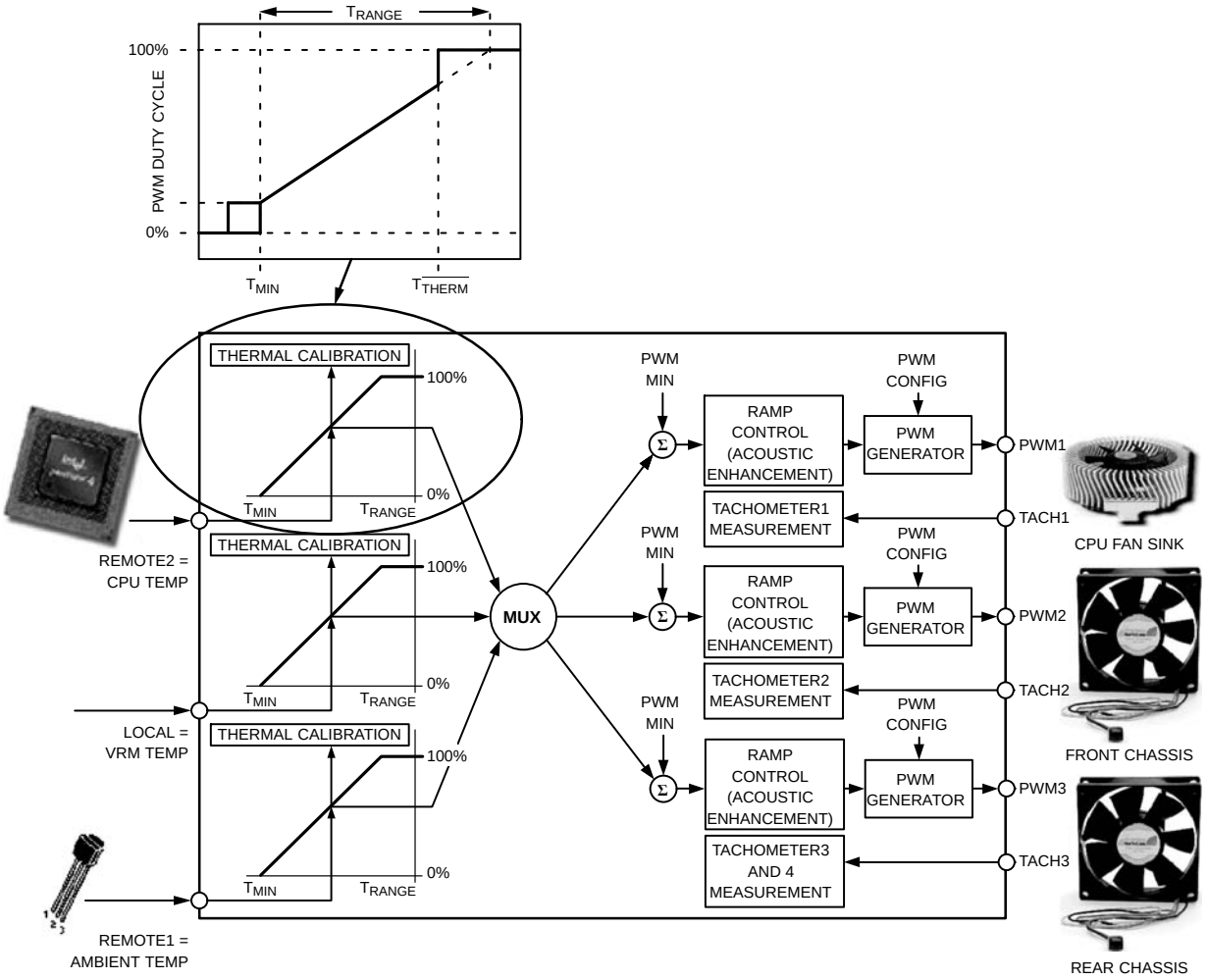



**Figure 63. The $T_{HYST}$ Value Applies to Fan On/Off Hysteresis and $\overline{THERM}$ Hysteresis**

### Enhanced Acoustics Register 1 (0x62)

Bit 7 (MIN3) = 0, PWM3 is off (0% PWM duty cycle) when temperature is below $T_{MIN} - T_{HYST}$.

Bit 7 (MIN3) = 1, PWM3 runs at PWM3 minimum duty cycle below $T_{MIN} - T_{HYST}$.

Bit 6 (MIN2) = 0, PWM2 is off (0% PWM duty cycle) when temperature is below $T_{MIN} - T_{HYST}$.

Bit 6 (MIN2) = 1, PWM2 runs at PWM2 minimum duty cycle below $T_{MIN} - T_{HYST}$.

Bit 5 (MIN1) = 0, PWM1 is off (0% PWM duty cycle) when temperature is below $T_{MIN} - T_{HYST}$.

Bit 5 (MIN1) = 1, PWM1 runs at PWM1 minimum duty cycle below $T_{MIN} - T_{HYST}$.

### Dynamic $T_{MIN}$ Control Mode

In addition to the automatic fan speed control mode described in the Automatic Fan Control Overview section, the ADT7473/ADT7473−1 has a mode that extends the basic automatic fan speed control loop. Dynamic $T_{MIN}$

control allows the ADT7473/ADT7473−1 to intelligently adapt the system's cooling solution for best system performance or lowest possible system acoustics, depending on user or design requirements. Use of dynamic $T_{MIN}$ control alleviates the need to design for worst-case conditions and significantly reduces system design and validation time.

### Designing for Worst-case Conditions

System design must always allow for worst-case conditions. In PC design, the worst-case conditions include, but are not limited to, the following:

### Worst-case Altitude

A computer can be operated at different altitudes. The altitude affects the relative air density, which alters the effectiveness of the fan cooling solution. For example, comparing 40°C air temperature at 10,000 feet to 20°C air temperature at sea level, relative air density is increased by 40%. This means that the fan can spin 40% slower and make less noise at sea level than at 10,000 feet while keeping the system at the same temperature at both locations.

### Worst-case Fan

Due to manufacturing tolerances, fan speeds in RPM are normally quoted with a tolerance of ±20%. The designer needs to assume that the fan RPM can be 20% below tolerance. This translates to reduced system airflow and elevated system temperature. Note that fans 20% out of tolerance can negatively impact system acoustics because they run faster and generate more noise.

### Worst-case Chassis Airflow

The same motherboard can be used in a number of different chassis configurations. The design of the chassis and the physical location of fans and components determine the system thermal characteristics. Moreover, for a given chassis, the addition of add-in cards, cables, or other system configuration options can alter the system airflow and reduce the effectiveness of the system cooling solution. The cooling solution can also be inadvertently altered by the end user. (For example, placing a computer against a wall can block the air ducts and reduce system airflow.)

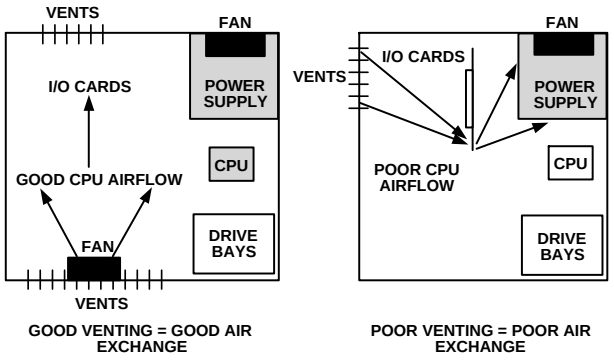



**Figure 64. Chassis Airflow Issues**

### Worst-case Processor Power Consumption

This data sheet maximum does not necessarily reflect the true processor power consumption. Designing for worst-case CPU power consumption can result in a processor becoming overcooled (generating excess system noise).

### Worst-case Peripheral Power Consumption

The tendency is to design to data sheet maximums for peripheral components, again overcooling the system.

### Worst-case Assembly

Every system is unique because of manufacturing variations. Heat sinks may be loose fitting or slightly misaligned. Too much or too little thermal grease might be used, or variations in application pressure for thermal interface material could affect the efficiency of the thermal solution. Accounting for manufacturing variations in every system is difficult; therefore, the system must be designed for the worst-case conditions.

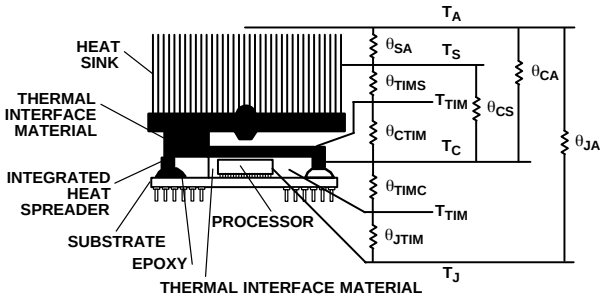



**Figure 65. Thermal Model**

Although a design usually accounts for worst-case conditions in all these cases, the actual system is almost never operated at worst-case conditions. The alternative to designing for the worst case is to use the dynamic $T_{MIN}$ control function.

### Dynamic $T_{MIN}$ Control Overview

Dynamic $T_{MIN}$ control mode builds on the basic automatic fan control loop by adjusting the $T_{MIN}$ value based on system performance and measured temperature. This is important because, instead of designing for the worst case, the system thermals can be defined as operating zones. The ADT7473/ADT7473−1 can self-adjust its fan control loop to maintain either an operating zone temperature or a system target temperature. For example, it can be specified that the ambient temperature in a system should be maintained at 50°C. If the temperature is below 50°C, the fans might not need to run, or might run very slowly. If the temperature is higher than 50°C, the fans need to throttle up.

The challenge presented by any thermal design is finding the right settings to suit the system's fan control solution. This can involve designing for the worst case, followed by weeks of system thermal characterization, and finally fan acoustic optimization (for psycho-acoustic reasons).

Getting the most benefit from the automatic fan control mode involves characterizing the system to find the best $T_{MIN}$ and $T_{RANGE}$ settings for the control loop, and the best $PWM_{MIN}$ value for the quietest fan speed setting. Using the ADT7473/ADT7473−1's dynamic $T_{MIN}$ control mode, however, shortens the characterization time and alleviates tweaking the control loop settings because the device can self-adjust during system operation.

Dynamic $T_{MIN}$ control mode is operated by specifying the operating zone temperatures required for the system. Associated with this control mode are three operating point registers, one for each temperature channel. This allows the system thermal solution to be broken down into distinct thermal zones. For example, CPU operating temperature is 70°C, VRM operating temperature is 80°C, and ambient operating temperature is 50°C. The ADT7473/ADT7473−1 dynamically alters the control solution to maintain each zone temperature as closely as possible to its target operating point.

**Table 48. OPERATING POINT REGISTERS**

| Register | Description | Default |
|---|---|---|
| 0x33 | Remote 1 Operating Point | 0xA4 (100°C) |
| 0x34 | Local Temp. Operating Point | 0xA4 (100°C) |
| 0x35 | Remote 2 Operating Point | 0xA4 (100°C) |

Figure 66 shows an overview of the parameters that affect the operation of the dynamic $T_{MIN}$ control loop.

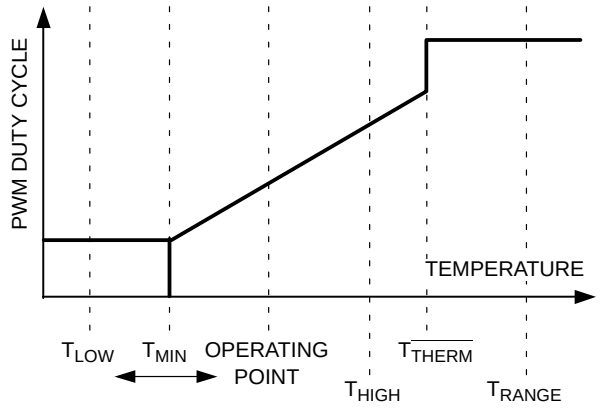



**Figure 66. Dynamic $T_{MIN}$ Control Loop**

Table 49 provides a brief description of each parameter.

**Table 49. $T_{MIN}$ CONTROL LOOP PARAMETERS**

| Parameter | Description |
|---|---|
| $T_{LOW}$ | If the temperature drops below the $T_{LOW}$ limit, an error flag is set in a status register and an $\overline{SMBALERT}$ interrupt can be generated. |
| $T_{HIGH}$ | If the temperature exceeds the $T_{HIGH}$ limit, an error flag is set in a status register and an $\overline{SMBALERT}$ interrupt can be generated. |
| $T_{MIN}$ | The temperature at which the fan turns on under automatic fan speed control. |
| Operating Point | The target temperature for a particular temperature zone. The ADT7473/ADT7473−1 attempts to maintain system temperature at about the operating point by adjusting the $T_{MIN}$ parameter of the control loop. |
| $T_{\overline{THERM}}$ | If the temperature exceeds this critical limit, the fans can be run at 100% for maximum cooling. |
| $T_{RANGE}$ | Programs the PWM duty cycle vs. temperature control slope. |

### Dynamic $T_{MIN}$ Control Programming

Because the dynamic $T_{MIN}$ control mode is a basic extension of the automatic fan control mode, program the automatic fan control mode parameters first, as described in the Step 1: Hardware Configuration section to the Step 8: $T_{HYST}$ for Temperature Channels section, then proceed with dynamic $T_{MIN}$ control mode programming.

### Step 9: Operating Points for Temperature Channels

The operating point for each temperature channel is the optimal temperature for that thermal zone. The hotter each zone is allowed to be, the quieter the system, because the fans are not required to run as fast. The ADT7473/ ADT7473−1 increases or decreases fan speeds as necessary to maintain the operating point temperature, allowing for system-to-system variation and removing the need for worst-case design. If a sensible operating point value is chosen, any $T_{MIN}$ value can be selected in the system characterization. If the $T_{MIN}$ value is too low, the fans run sooner than required, and the temperature is below the operating point. In response, the ADT7473/ADT7473−1 increases $T_{MIN}$ to keep the fans off longer and to allow the temperature zone to get closer to the operating point. Likewise, too high a $T_{MIN}$ value causes the operating point to be exceeded, and in turn, the ADT7473/ADT7473−1 reduces $T_{MIN}$ to turn the fans on sooner to cool the system.

### Programming Operating Point Registers

There are three operating point registers, one for each temperature channel. These 8-bit registers allow the operating point temperatures to be programmed with 1°C resolution.

**Table 50. OPERATING POINT REGISTERS**

| Register | Description | Default |
|---|---|---|
| 0x33 | Remote 1 Operating Point | 0xA4 (100°C) |
| 0x34 | Local Operating Point | 0xA4 (100°C) |
| 0x35 | Remote 2 Operating Point | 0xA4 (100°C) |

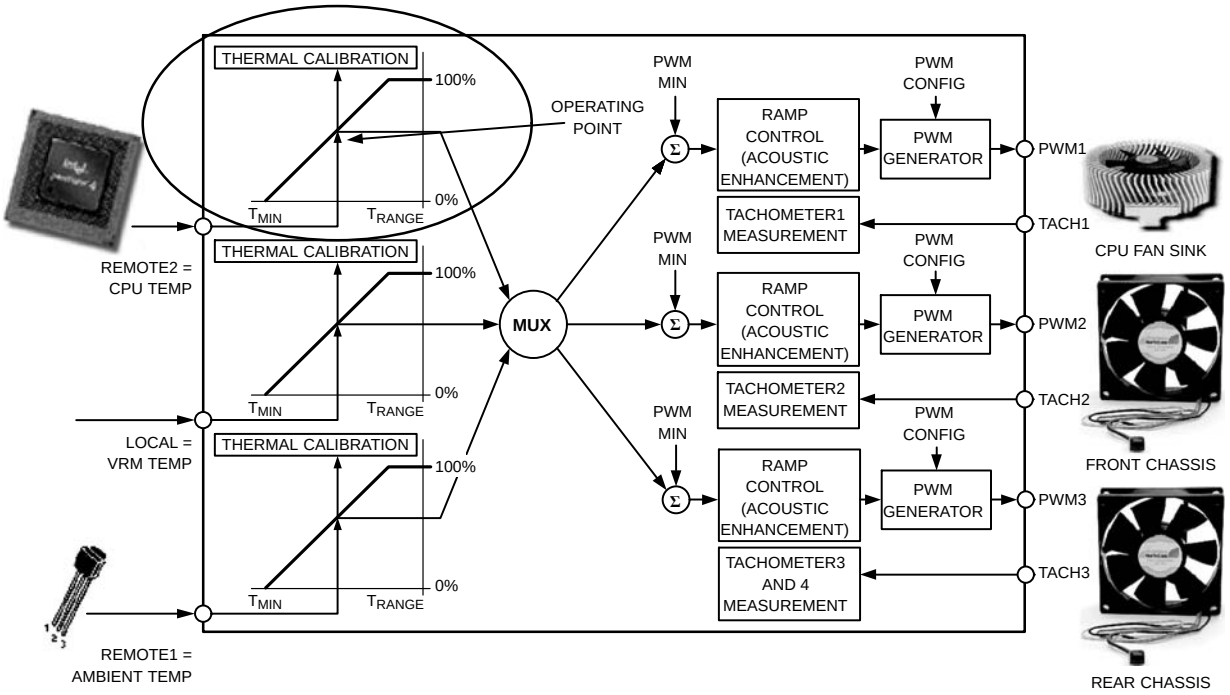



**Figure 67. Operating Point Value Dynamically Adjusts Automatic Fan Control Settings**

### Step 10: High and Low Limits for Temperature Channels

The low limit defines the temperature at which the $T_{MIN}$ value starts to be increased, if temperature falls below this value. This has the net effect of reducing the fan speed, allowing the system to get hotter. An interrupt can be generated when the temperature drops below the low limit.

The high limit defines the temperature at which the $T_{MIN}$ value starts to be reduced, if temperature increases above this value. This has the net effect of increasing fan speed to cool down the system. An interrupt can be generated when the temperature rises above the high limit.

### Programming High and Low Limits

There are six limit registers; a high limit and low limit are associated with each temperature channel. These 8-bit registers allow the high and low limit temperatures to be programmed with 1°C resolution.

**Table 51. TEMPERATURE LIMIT REGISTERS**

| Register | Description | Default |
|---|---|---|
| 0x4E | Remote 1 Temperature Low Limit | 0x01 |
| 0x4F | Remote 1 Temperature High Limit | 0x7F |
| 0x50 | Local Temperature Low Limit | 0x01 |
| 0x51 | Local Temperature High Limit | 0x7F |
| 0x52 | Remote 2 Temperature Low Limit | 0x01 |
| 0x53 | Remote 2 Temperature High Limit | 0x7F |

### How Dynamic $T_{MIN}$ Control Works

The basic premise is as follows:

1. Set the target temperature for the temperature zone, which could be, for example, the Remote 1 thermal diode. This value is programmed to the Remote 1 operating temperature register.

2. As the temperature in that zone (Remote 1 temperature) rises toward and exceeds the operating point temperature, $T_{MIN}$ is reduced, and the fan speed increases.
3. As the temperature drops below the operating point temperature, $T_{MIN}$ is increased, and the fan speed is reduced.

However, the loop operation is not as simple as described in these steps. A number of conditions govern the situations in which $T_{MIN}$ can increase or decrease.

### Short Cycle and Long Cycle

The ADT7473/ADT7473−1 implements two loops: a short cycle and a long cycle. The short cycle takes place every n monitoring cycles. The long cycle takes place every 2n monitoring cycles. The value of n is programmable for each temperature channel. The bits are located at the following register locations:

Remote 1 = CYR1 = Bits <2:0> of Dynamic $T_{MIN}$ Control Register 2 (0x37).

Local = CYL = Bits <5:3> of Dynamic $T_{MIN}$ Control Register 2 (0x37).

Remote 2 = CYR2 = Bits <7:6> of Dynamic $T_{MIN}$ Control Register 2 (0x37) and Bit 0 of Dynamic $T_{MIN}$ Control Register 1 (0x36).

**Table 52. CYCLE BIT ASSIGNMENTS**

| Code | Short Cycle | Secs | Long Cycle | Secs |
|---|---|---|---|---|
| 000 | 8 cycles | 1 sec | 16 cycles | 2 sec |
| 001 | 16 cycles | 2 sec | 32 cycles | 4 sec |
| 010 | 32 cycles | 4 sec | 64 cycles | 8 sec |
| 011 | 64 cycles | 8 sec | 128 cycles | 16 sec |
| 100 | 128 cycles | 16 sec | 256 cycles | 32 sec |
| 101 | 256 cycles | 32 sec | 512 cycles | 64 sec |
| 110 | 512 cycles | 64 sec | 1024 cycles | 128 sec |
| 111 | 1024 cycles | 128 sec | 2048 cycles | 256 sec |

Care should be taken when choosing the cycle time. A long cycle time means that $T_{MIN}$ is updated less often. If your system has very fast temperature transients, the dynamic $T_{MIN}$ control loop is always lagging. If a cycle time is chosen that is too fast, the full benefit of changing $T_{MIN}$ might not be realized and needs to change again on the next cycle; in effect, it is overshooting. It is necessary to carry out some calibration to identify the most suitable response time.

Figure 68 shows the steps taken during the short cycle.

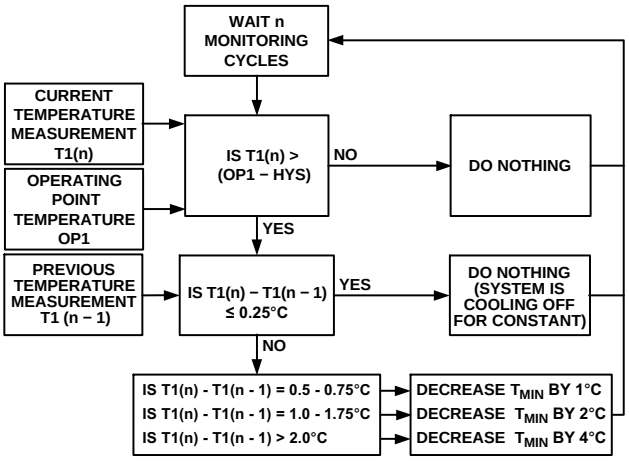



**Figure 68. Short Cycle Steps**

Figure 69 shows the steps taken during the long cycle.

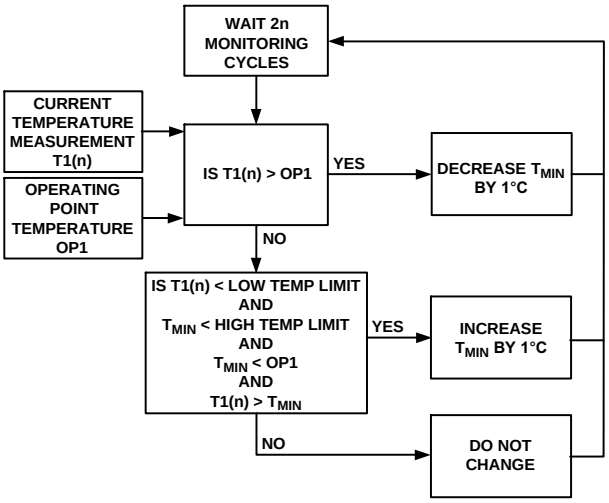



**Figure 69. Long Cycle Steps**

The following examples illustrate some of the circumstances that might cause $T_{MIN}$ to increase, decrease, or stay the same.

### Example 1: Normal Operation − No $T_{MIN}$ Adjustment

- If measured temperature never exceeds the programmed operating point minus the hysteresis temperature, then $T_{MIN}$ is not adjusted; that is, it remains at its current setting.
- If measured temperature never drops below the low temperature limit, then $T_{MIN}$ is not adjusted.

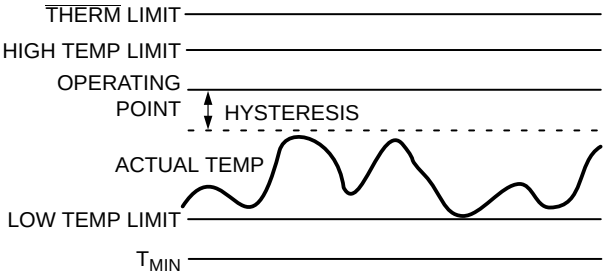



**Figure 70. Temperature Between the Operating Point and the Low Temperature Limit**

Because neither the operating point minus the hysteresis temperature nor the low temperature limit has been exceeded, the $T_{MIN}$ value is not adjusted, and the fan runs at a speed determined by the fixed $T_{MIN}$ and $T_{RANGE}$ values defined in the automatic fan speed control mode.

### Example 2: Operating Point Exceeded − $T_{MIN}$ Reduced

When the measured temperature is below the operating point temperature minus the hysteresis, $T_{MIN}$ remains the same.

Once the temperature exceeds the operating temperature minus the hysteresis (OP − Hyst), $T_{MIN}$ starts to decrease. This occurs during the short cycle (see Figure 68). The rate at which $T_{MIN}$ decreases depends on the programmed value of n. It also depends on how much the temperature has increased between this monitoring cycle and the last monitoring cycle; that is, if the temperature has increased by 1°C, then $T_{MIN}$ is reduced by 2°C. Decreasing $T_{MIN}$ has the effect of increasing the fan speed, thus providing more cooling to the system.

If the temperature slowly increases only in the range (OP − Hyst), that is, ≤0.25°C per short monitoring cycle, then $T_{MIN}$ does not decrease. This allows small changes in temperature in the desired operating zone without changing $T_{MIN}$. The long cycle makes no change to $T_{MIN}$ in the temperature range (OP − Hyst) because the temperature has not exceeded the operating temperature.

Once the temperature exceeds the operating temperature, the long cycle causes $T_{MIN}$ to be reduced by 1°C every long cycle while the temperature remains above the operating temperature. This takes place in addition to the decrease in $T_{MIN}$ that occurs due to the short cycle. In Figure 71, because the temperature is increasing at a rate ≤0.25°C per short cycle, no reduction in $T_{MIN}$ takes place during the short cycle.

Once the temperature falls below the operating temperature, $T_{MIN}$ stays the same. Even when the temperature starts to increase slowly, $T_{MIN}$ stays the same because the temperature increases at a rate ≤0.25°C per cycle.

### Example 3: Increase $T_{MIN}$ Cycle

When the temperature drops below the low temperature limit, $T_{MIN}$ can increase in the long cycle. Increasing $T_{MIN}$ has the effect of running the fan slower and, therefore, quieter. The long cycle diagram in Figure 69 shows the conditions required for $T_{MIN}$ to increase. A quick summary of those conditions and the reasons they need to be true follows.

$T_{MIN}$ can increase if:

- The measured temperature falls below the low temperature limit. This means the user must choose the low limit carefully. It should not be so low that the temperature never falls below it because $T_{MIN}$ would never increase, and the fans would run faster than necessary.
- $T_{MIN}$ is below the high temperature limit. $T_{MIN}$ is never allowed to increase above the high temperature limit. As a result, the high limit should be sensibly chosen because it determines how high $T_{MIN}$ can go.
- $T_{MIN}$ is below the operating point temperature. $T_{MIN}$ should never be allowed to increase above the operating point temperature because the fans would not switch on until the temperature rose above the operating point.
- The temperature is above $T_{MIN}$. The dynamic $T_{MIN}$ control is turned off below $T_{MIN}$.

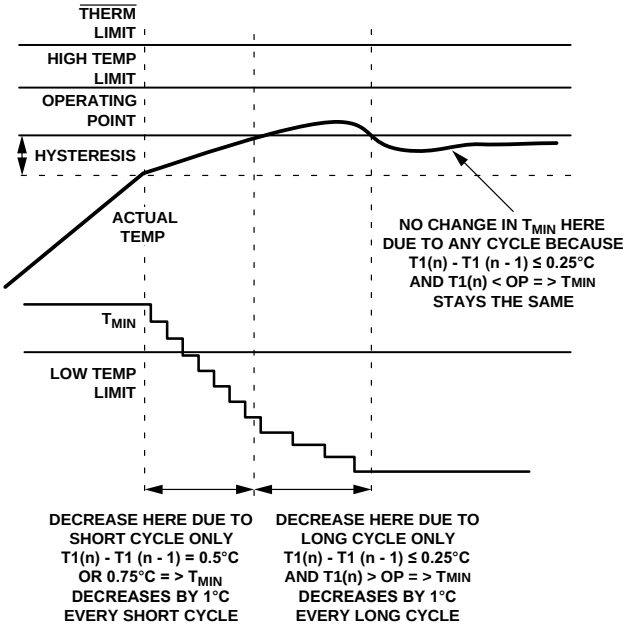



**Figure 71. Effect of Exceeding Operating Point Minus Hysteresis Temperature**

Figure 72 shows how $T_{MIN}$ increases when the current temperature is above $T_{MIN}$ and below the low temperature limit, and $T_{MIN}$ is below the high temperature limit and below the operating point. Once the temperature rises above the low temperature limit, $T_{MIN}$ stays the same.

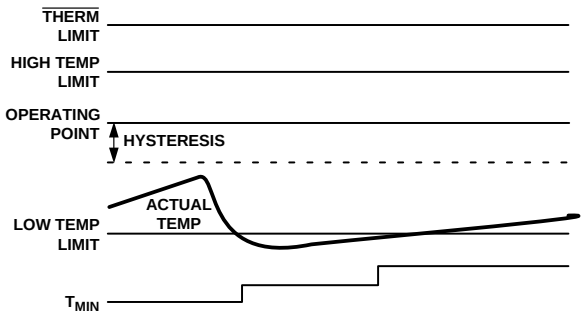



**Figure 72. Increasing $T_{MIN}$ for Quieter Operation**

### Example 4: Preventing $T_{MIN}$ from Reaching Full Scale

Because $T_{MIN}$ is dynamically adjusted, it is undesirable for $T_{MIN}$ to reach full scale (127°C) because the fan would never switch on. As a result, $T_{MIN}$ is allowed to vary only within a specified range:

- The lowest possible value for $T_{MIN}$ is −127°C (twos complement mode) or −64°C (Offset 64 mode).
- $T_{MIN}$ cannot exceed the high temperature limit.
- If the temperature is below $T_{MIN}$, the fan is switched off or runs at minimum speed and dynamic $T_{MIN}$ control is disabled.

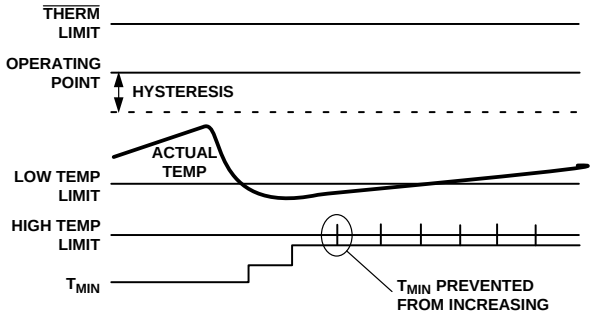



**Figure 73. $T_{MIN}$ Adjustments Limited by the High Temperature Limit**

### Step 11: Monitoring $\overline{\text{THERM}}$

Using the operating point limit ensures that the dynamic $T_{MIN}$ control mode operates in the best possible acoustic position while ensuring that the temperature never exceeds the maximum operating temperature. Using the operating point limit allows $T_{MIN}$ to be independent of system-level issues because of its self-corrective nature. In PC design, the operating point for the chassis is usually the worst-case internal chassis temperature.

The optimal operating point for the processor is determined by monitoring the thermal monitor in the Intel® Pentium® 4 processor. To do this, the $\overline{\text{PROCHOT}}$ output of the Pentium® 4 is connected to the $\overline{\text{THERM}}$ input of the ADT7473/ADT7473−1.

The operating point for the processor can be determined by allowing the current temperature to be copied to the operating point register when the $\overline{\text{PROCHOT}}$ output pulls the $\overline{\text{THERM}}$ input low on the ADT7473/ADT7473−1. This gives the maximum temperature at which the Pentium® 4 can run before clock modulation occurs.

### Enabling the $\overline{\text{THERM}}$ Trip Point as the Operating Point

Bits <4:2> of Dynamic $T_{MIN}$ Control Register 1 (0x36) enable/disable $\overline{\text{THERM}}$ monitoring to program the operating point.

### Dynamic $T_{MIN}$ Control Register 1 (0x36)

Bit <4> PHTR2 = 1, copies the Remote 2 current temperature to the Remote 2 operating point register, if $\overline{\text{THERM}}$ is asserted. The operating point contains the temperature at which $\overline{\text{THERM}}$ is asserted. This allows the system to run as quietly as possible without affecting system performance.

PHTR2 = 0, ignores any $\overline{\text{THERM}}$ assertions. The Remote 2 operating point register reflects its programmed value.

Bit <3> PHTL = 1, copies the local current temperature to the local temperature operating point register if $\overline{\text{THERM}}$ is asserted. The operating point contains the temperature at which $\overline{\text{THERM}}$ is asserted. This allows the system to run as quietly as possible without affecting system performance.

PHTL = 0, ignores any $\overline{\text{THERM}}$ assertions. The local temperature operating point register reflects its programmed value.

Bit <2> PHTR1 = 1, copies the Remote 1 current temperature to the Remote 1 operating point register if $\overline{\text{THERM}}$ is asserted. The operating point contains the temperature at which $\overline{\text{THERM}}$ is asserted. This allows the system to run as quietly as possible without affecting system performance.

PHTR1 = 0, ignores any $\overline{\text{THERM}}$ assertions. The Remote 1 operating point register reflects its programmed value.

### Enabling Dynamic $T_{MIN}$ Control Mode

Bits <7:5> of the Dynamic $T_{MIN}$ Control Register 1 (0x36) enable/disable dynamic $T_{MIN}$ control on the temperature channels.

### Dynamic $_{TMIN}$ Control Register 1 (0x36)

Bit <7> R2T = 1, enables dynamic $T_{MIN}$ control on the Remote 2 temperature channel. The chosen $T_{MIN}$ value is dynamically adjusted based on the current temperature, operating point, and high and low limits for this zone.

R2T = 0, disables dynamic $T_{MIN}$ control. The $T_{MIN}$ value chosen is not adjusted and the channel behaves as described in the Automatic Fan Control Overview section.

Bit <6> LT = 1, enables dynamic $T_{MIN}$ control on the local temperature channel. The chosen $T_{MIN}$ value is dynamically adjusted based on the current temperature, operating point, and high and low limits for this zone.

LT = 0, disables dynamic $T_{MIN}$ control. The $T_{MIN}$ value chosen is not adjusted and the channel behaves as described in the Automatic Fan Control Overview section.

Bit <5> R1T = 1, enables dynamic $T_{MIN}$ control on the Remote 1 temperature channel. The chosen $T_{MIN}$ value is dynamically adjusted based on the current temperature, operating point, and high and low limits for this zone.

R1T = 0, disables dynamic $T_{MIN}$ control. The $T_{MIN}$ value chosen is not adjusted, and the channel behaves as described in the Automatic Fan Control Overview section.

### Enhancing System Acoustics

Automatic fan speed control mode reacts instantaneously to changes in temperature; that is, the PWM duty cycle responds immediately to temperature change. Any impulses in temperature can cause an impulse in fan noise. For psycho-acoustic reasons, the ADT7473/ADT7473−1 can prevent the PWM output from reacting instantaneously to temperature changes. Enhanced acoustic mode controls the maximum change in PWM duty cycle at a given time. The objective is to prevent the fan from cycling up and down, annoying the user.

### Acoustic Enhancement Mode Overview

Figure 74 gives a top-level overview of the automatic fan control circuitry on the ADT7473/ADT7473−1 and shows where acoustic enhancement fits in. Acoustic enhancement is intended as a post design tweak made by a system or mechanical engineer evaluating best settings for the system. Having determined the optimal settings for the thermal solution, the engineer can adjust the system acoustics. The goal is to implement a system that is acoustically pleasing without causing user annoyance due to fan cycling. It is important to realize that although a system might pass an acoustic noise requirement specification (for example, 36 dB), if the fan is annoying, it fails the consumer test.

### Approaches to System Acoustic Enhancement

There are two different approaches to implementing system acoustic enhancement: temperature-centric and fan-centric.

The temperature-centric approach involves smoothing transient temperatures as they are measured by a temperature source (for example, Remote 1 temperature). The temperature values used to calculate the PWM duty cycle values are smoothed, reducing fan speed variation. However, this approach causes an inherent delay in updating fan speed and causes the thermal characteristics of the system to change. It also causes the system fans to stay on longer than necessary because the fan's reaction is merely delayed. The user has no control over noise from different fans driven by the same temperature source. Consider, for example, a system in which control of a CPU cooler fan (on PWM1) and a chassis fan (on PWM2) use Remote 1 temperature. Because the Remote 1 temperature is smoothed, both fans are updated at exactly the same rate. If the chassis fan is much louder than the CPU fan, there is no way to improve its acoustics without changing the thermal solution of the CPU cooling fan.

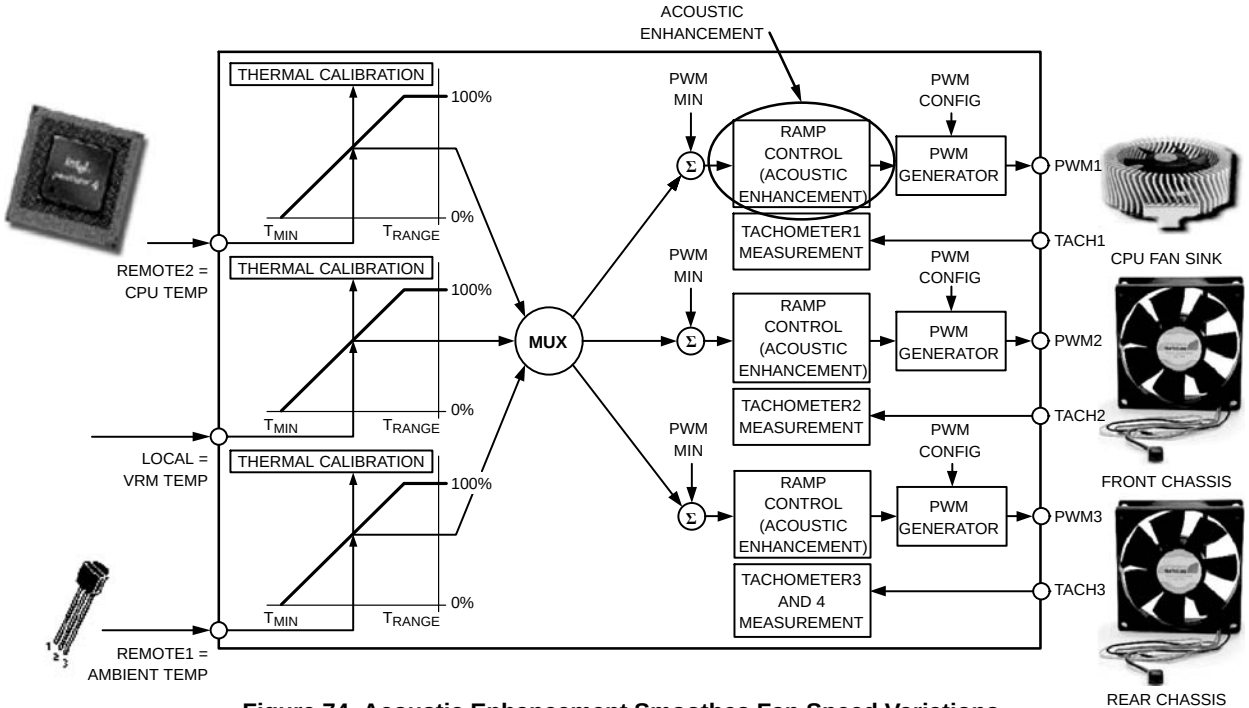



**Figure 74. Acoustic Enhancement Smoothes Fan Speed Variations Under Automatic Fan Speed Control**

The fan-centric approach to system acoustic enhancement controls the PWM duty cycle, driving the fan at a fixed rate (for example, 6%). Each time the PWM duty cycle is updated, it is incremented by a fixed 6%. As a result, the fan ramps smoothly to its newly calculated speed. If the temperature starts to drop, the PWM duty cycle immediately decreases by 6% at every update. Therefore, the fan ramps smoothly up or down without inherent system delay. Consider, for example, controlling the same CPU cooler fan (on PWM1) and chassis fan (on PWM2) using Remote 1 temperature. The $T_{MIN}$ and $T_{RANGE}$ settings have already been defined in automatic fan speed control mode, that is, thermal characterization of the control loop has been optimized. Here, the chassis fan is noisier than the CPU cooling fan. Using the fan-centric approach, PWM2 can be placed into acoustic enhancement mode independently of PWM1. The acoustics of the chassis fan can, therefore, be adjusted without affecting the acoustic behavior of the CPU cooling fan, even though both fans are controlled by Remote 1 temperature. The fan-centric approach is how acoustic enhancement works on the ADT7473/ ADT7473−1.

### Enabling Acoustic Enhancement for Each PWM Output

#### Enhanced acoustics Register 1 (0x62)

Bit 3 = 1, enables acoustic enhancement on PWM1 output

#### Enhanced acoustics Register 2 (0x63)

Bit 7 = 1, enables acoustic enhancement on PWM2 output

Bit 3 = 1, enables acoustic enhancement on PWM3 output

### Effect of Ramp Rate on Enhanced Acoustics Mode

The PWM signal driving the fan has a period, T, given by the PWM drive frequency, f, because T = 1/f. For a given PWM period, T, the PWM period is subdivided into 255 equal time slots. One time slot corresponds to the smallest possible increment in the PWM duty cycle. A PWM signal of 33% duty cycle is, therefore, high for $1/3 \times 255$ time slots and low for $2/3 \times 255$ time slots. Therefore, a 33% PWM duty cycle corresponds to a signal that is high for 85 time slots and low for 170 time slots.

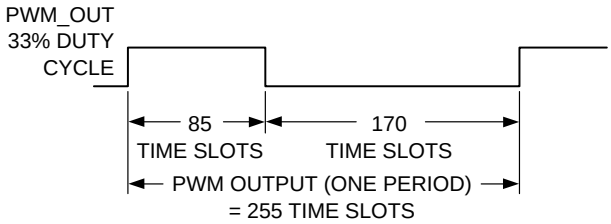



**Figure 75. 33% PWM Duty Cycle Represented in Time Slots**

The ramp rates in the enhanced acoustics mode are selectable from the values 1, 2, 3, 5, 8, 12, 24, and 48. The ramp rates are discrete time slots. For example, if the ramp rate is 8, then eight time slots are added to the PWM high duty cycle each time the PWM duty cycle needs to be increased. If the PWM duty cycle value needs to be decreased, it is decreased by eight time slots. Figure 76 shows how the enhanced acoustics mode algorithm operates.

The enhanced acoustics mode algorithm calculates a new PWM duty cycle based on the temperature measured. If the new PWM duty cycle value is greater than the previous PWM value, then the previous PWM duty cycle value is incremented by either 1, 2, 3, 5, 8, 12, 24, or 48 time slots, depending on the settings of the enhanced acoustics registers. If the new PWM duty cycle value is less than the previous PWM value, the previous PWM duty cycle is decremented by 1, 2, 3, 5, 8, 12, 24, or 48 time slots. Each time the PWM duty cycle is incremented or decremented, its value is stored as the previous PWM duty cycle for the next comparison. A ramp rate of 1 corresponds to one time slot, which is 1/255 of the PWM period. In enhanced acoustics mode, incrementing or decrementing by 1 changes the PWM output by $1/255 \times 100\%$.

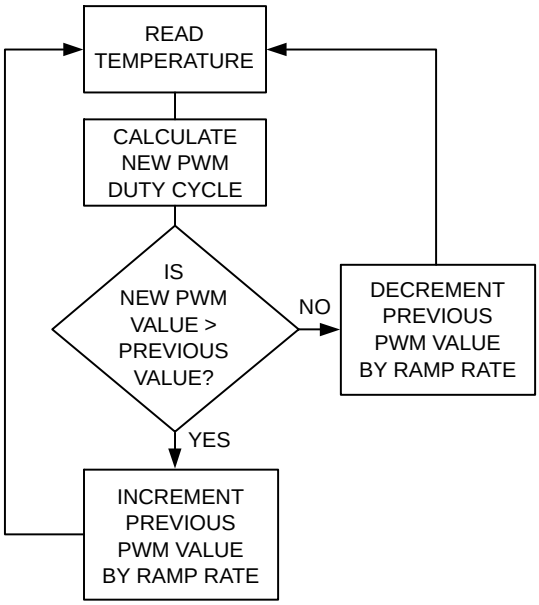



**Figure 76. Enhanced Acoustics Algorithm**

### Step 12: Ramp Rate for Acoustic Enhancement

The optimal ramp rate for acoustic enhancement can be found through system characterization after the thermal optimization has been finished. The effect of each ramp rate should be logged, if possible, to determine the best setting for a given solution.

#### Enhanced Acoustics Register 1 (0x62)

Bits <2:0> ACOU, select the ramp rate for PWM1.

000 = 1 time slot = 35 sec
001 = 2 time slots = 17.6 sec
010 = 3 time slots = 11.8 sec
011 = 5 time slots = 7 sec
100 = 8 time slots = 4.4 sec
101 = 12 time slots = 3 sec
110 = 24 time slots = 1.6 sec
111 = 48 time slots = 0.8 sec

**Enhanced Acoustics Register 2 (0x63)**

Bits <2:0> ACOU3, select the ramp rate for PWM3.

000 = 1 time slot = 35 sec
001 = 2 time slots = 17.6 sec
010 = 3 time slots = 11.8 sec
011 = 5 time slots = 7 sec
100 = 8 time slots = 4.4 sec
101 = 12 time slots = 3 sec
110 = 24 time slots = 1.6 sec
111 = 48 time slots = 0.8 sec

Bits <6:4> ACOU2, select the ramp rate for PWM2.

000 = 1 time slot = 35 sec
001 = 2 time slots = 17.6 sec
010 = 3 time slots = 11.8 sec
011 = 5 time slots = 7 sec
100 = 8 time slots = 4.4 sec
101 = 12 time slots = 3 sec
110 = 24 time slots = 1.6 sec
111 = 48 time slots = 0.8 sec

Another way to view the ramp rates is to measure the time it takes for the PWM output to ramp up from 0% to 100% duty cycle for an instantaneous change in temperature. This can be tested by putting the ADT7473/ADT7473−1 into manual mode and changing the PWM output from 0% to 100% PWM duty cycle. The PWM output takes 35 seconds to reach 100% when a ramp rate of 1 time slot is selected.

Figure 77 shows remote temperature plotted against PWM duty cycle for enhanced acoustics mode. The ramp rate is set to 48, which corresponds to the fastest ramp rate. Assume that a new temperature reading is available every 115 ms. With these settings, it takes approximately 0.76 seconds to go from 33% duty cycle to 100% duty cycle (full speed). Even though the temperature increases very rapidly, the fan ramps up to full speed gradually.

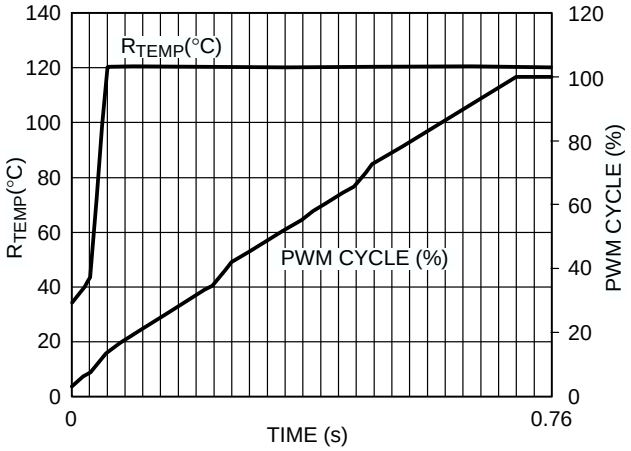



**Figure 77. Enhanced Acoustics Mode with Ramp Rate = 48**

Figure 78 shows how changing the ramp rate from 48 to 8 affects the control loop. The overall response of the fan is slower. Because the ramp rate is reduced, it takes longer for the fan to achieve full running speed. In this case, it takes approximately 4.4 seconds for the fan to reach full speed.

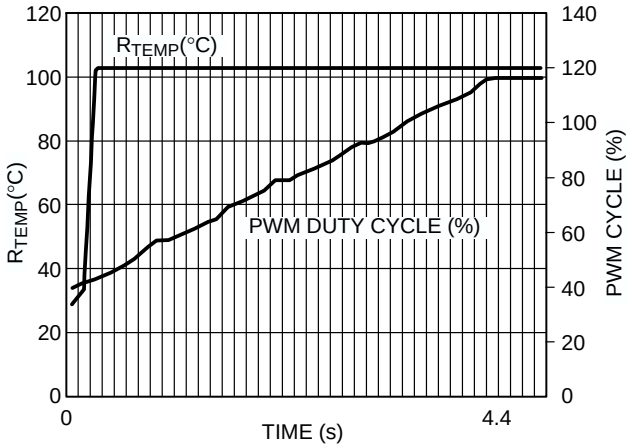



**Figure 78. Enhanced Acoustics Mode with Ramp Rate = 8**

Figure 79 shows the PWM output response for a ramp rate of 2. In this instance, the fan takes about 17.6 seconds to reach full running speed.

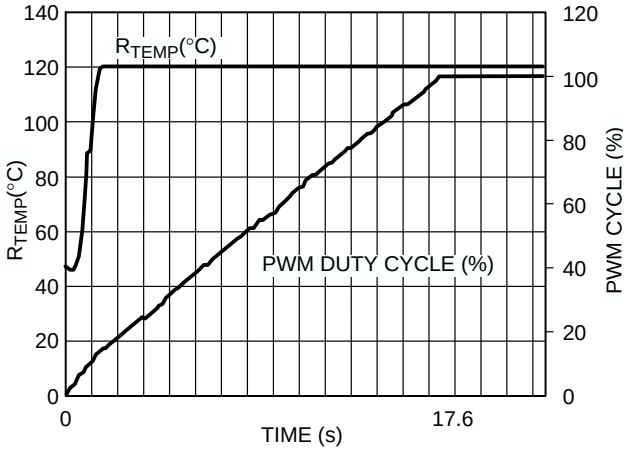



**Figure 79. Enhanced Acoustics Mode with Ramp Rate = 2**

Figure 80 shows how the control loop reacts to temperature with the slowest ramp rate. The ramp rate is set to 1, while all other control parameters remain the same. With the slowest ramp rate selected, it takes 35 seconds for the fan to reach full speed.

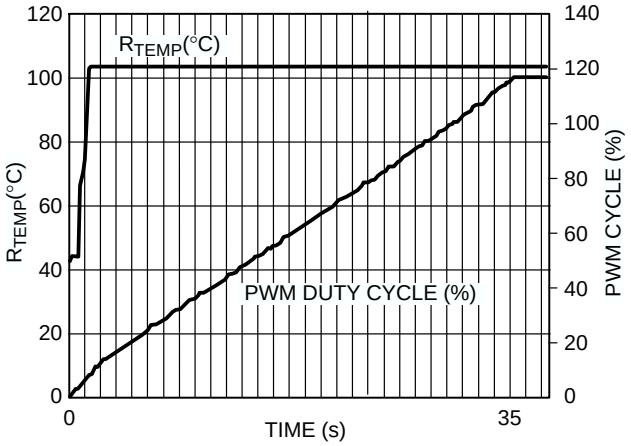



**Figure 80. Enhanced Acoustics Mode with Ramp Rate = 1**

As Figure 77 to Figure 80 show, the rate at which the fan reacts to temperature change is dependent on the ramp rate selected in the enhanced acoustics registers. The higher the ramp rate, the faster the fan reaches the newly calculated fan speed.

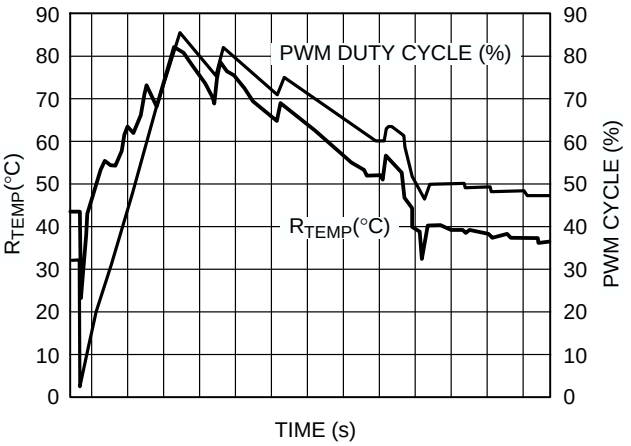



**Figure 81. How Fan Reacts to Temperature Variations in Enhanced Acoustics Mode**

Figure 81 shows the behavior of the PWM output as temperature varies. As the temperature increases, the fan speed ramps up. Small drops in temperature do not affect the ramp-up function because the newly calculated fan speed is still higher than the previous PWM value. Enhanced acoustics mode allows the PWM output to be made less sensitive to temperature variations. This is dependent on the ramp rate selected and programmed into the enhanced acoustics registers.

### Slower Ramp Rates

The ADT7473/ADT7473−1 can be programmed for much longer ramp times by slowing the ramp rates. Each ramp rate can be slowed by a factor of 4.

#### PWM1 Configuration Register (0x5C)

Bit <3> SLOW, 1 slows the ramp rate for PWM1 by 4.

#### PWM2 Configuration Register (0x5D)

Bit <3> SLOW, 1 slows the ramp rate for PWM2 by 4.

#### PWM3 Configuration Register (0x5E)

Bit <3> SLOW, 1 slows the ramp rate for PWM3 by 4.

The following sections list the ramp-up times when the SLOW bit is set for each PWM output.

#### Enhanced Acoustics Register 1 (0x62)

Bits <2:0> ACOU, select the ramp rate for PWM1.

000 = 140 sec
001 = 70.4 sec
010 = 47.2 sec
011 = 28 sec
100 = 17.6 sec
101 = 12 sec
110 = 6.4 sec
111 = 3.2 sec

#### Enhanced Acoustics Register 2 (0x63)

Bits <2:0> ACOU3, select the ramp rate for PWM3.

000 = 140 sec
001 = 70.4 sec
010 = 47.2 sec
011 = 28 sec
100 = 17.6 sec
101 = 12 sec
110 = 6.4 sec
111 = 3.2 sec

Bits <6:4> ACOU2, select the ramp rate for PWM2.

000 = 140 sec
001 = 70.4 sec
010 = 47.2 sec
011 = 28 sec
100 = 17.6 sec
101 = 12 sec
110 = 6.4 sec
111 = 3.2 sec

## Register Tables

### Table 53. ADT7473/ADT7473−1 REGISTERS

| Addr | R/W | Desc | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | De-fault | Lock-able |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x21 | R | $V_{CCP}$ Reading | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 0x00 | − |
| 0x22 | R | $V_{CC}$ Reading | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 0x00 | − |
| 0x25 | R | Remote 1 Temp. | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 0x01 | − |
| 0x26 | R | Local Temp. | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 0x01 | − |
| 0x27 | R | Remote 2 Temp. | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 0x01 | − |
| 0x28 | R | TACH1 Low Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x00 | − |
| 0x29 | R | TACH1 High Byte | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 0x00 | − |
| 0x2A | R | TACH2 Low Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x00 | − |
| 0x2B | R | TACH2 High Byte | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 0x00 | − |
| 0x2C | R | TACH3 Low Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x00 | − |
| 0x2D | R | TACH3 High Byte | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 0x00 | − |
| 0x2E | R | TACH4 Low Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x00 | − |
| 0x2F | R | TACH4 High Byte | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 0x00 | − |
| 0x30 | R/W | PWM1 Current Duty Cycle | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x00/0 xFF | − |
| 0x31 | R/W | PWM2 Current Duty Cycle | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x00/0 xFF | − |
| 0x32 | R/W | PWM3 Current Duty Cycle | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x00/0 xFF | − |
| 0x33 | R/W | Remote 1 Operating Point | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0xA4 | Yes |
| 0x34 | R/W | Local Temp Operating Point | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0xA4 | Yes |
| 0x35 | R/W | Remote 2 Operating Point | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0xA4 | Yes |
| 0x36 | R/W | Dynamic $T_{MIN}$ Control Reg. 1 | R2T | LT | R1T | PHTR2 | PHTL | PHTR1 | $V_{CCP}$LO | CYR2 | 0x00 | Yes |
| 0x37 | R/W | Dynamic $T_{MIN}$ Control Reg. 2 | CYR2 | CYR2 | CYL | CYL | CYL | CYR1 | CYR1 | CYR1 | 0x00 | Yes |
| 0x38 | R/W | PWM1 Max Duty Cycle | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0xFF | − |
| 0x39 | R/W | PWM2 Max Duty Cycle | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0xFF | − |
| 0x3A | R/W | PWM3 Max Duty Cycle | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0xFF | − |

**Table 53. ADT7473/ADT7473−1 REGISTERS** (continued)

| Addr | R/W | Desc | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | De-fault | Lock-able |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x3D | R | Device ID Register | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x73 | − |
| 0x3E | R | Company ID Number | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x41 | − |
| 0x3F | R | Revision ID Register | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x68/0 x69 | − |
| 0x40 | R/W | Config. Register 1 | ADT7473: RES ADT7473−1: Latch Reset | TODIS | FSPDIS | Vx1 | FSPD | RDY | LOCK | STRT | 0x01 | Yes |
| 0x41 | R | Interrupt Status Register 1 | OOL | R2T | LT | R1T | RES | $V_{CC}$ | $V_{CCP}$ | RES | 0x00 | − |
| 0x42 | R | Interrupt Status Register 2 | D2 | D1 | F4P | FAN3 | FAN2 | FAN1 | OVT | ADT7473: RES ADT7473−1: $\overline{\text{THERM}}$ Limit Latch | 0x00 | − |
| 0x46 | R/W | $V_{CCP}$ Low Limit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x00 | − |
| 0x47 | R/W | $V_{CCP}$ High Limit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0xFF | − |
| 0x48 | R/W | $V_{CC}$ Low Limit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x00 | − |
| 0x49 | R/W | $V_{CC}$ High Limit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0xFF | − |
| 0x4E | R/W | Remote 1 Temp. Low Limit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x01 | − |
| 0x4F | R/W | Remote 1 Temp. High Limit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0xFF | − |
| 0x50 | R/W | Local Temp. Low Limit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x01 | − |
| 0x51 | R/W | Local Temp. High Limit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0xFF | − |
| 0x52 | R/W | Remote 2 Temp. Low Limit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x01 | − |
| 0x53 | R/W | Remote 2 Temp. High Limit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0xFF | − |
| 0x54 | R/W | TACH1 Minimum Low Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0xFF | − |
| 0x55 | R/W | TACH1 Minimum High Byte | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 0xFF | − |
| 0x56 | R/W | TACH2 Minimum Low Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0xFF | − |
| 0x57 | R/W | TACH2 Minimum High Byte | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 0xFF | − |
| 0x58 | R/W | TACH3 Minimum Low Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0xFF | − |
| 0x59 | R/W | TACH3 Minimum High Byte | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 0xFF | − |

**Table 53. ADT7473/ADT7473−1 REGISTERS** (continued)

| Addr | R/W | Desc | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | De-fault | Lock-able |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x5A | R/W | TACH4 Minimum Low Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0xFF | − |
| 0x5B | R/W | TACH4 Minimum High Byte | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 0xFF | − |
| 0x5C | R/W | PWM1 Config. Register | BHVR | BHVR | BHVR | INV | SLOW | SPIN | SPIN | SPIN | 0x82/0 x62 | Yes |
| 0x5D | R/W | PWM2 Config. Register | BHVR | BHVR | BHVR | INV | SLOW | SPIN | SPIN | SPIN | 0x82/0 x62 | Yes |
| 0x5E | R/W | PWM3 Config. Register | BHVR | BHVR | BHVR | INV | SLOW | SPIN | SPIN | SPIN | 0x82/0 x62 | Yes |
| 0x5F | R/W | Remote 1 $T_{RANGE}$/ PWM 1 Frequency | RANGE | RANGE | RANGE | RANGE | HF/LF Fan 1 | FREQ | FREQ | FREQ | 0xCC | Yes |
| 0x60 | R/W | Local $T_{RANGE}$/ PWM 2 Frequency | RANGE | RANGE | RANGE | RANGE | HF/LF Fan 2 | FREQ | FREQ | FREQ | 0xCC | Yes |
| 0x61 | R/W | Remote 2 $T_{RANGE}$/ PWM 3 Frequency | RANGE | RANGE | RANGE | RANGE | HF/LF Fan 3 | FREQ | FREQ | FREQ | 0xCC | Yes |
| 0x62 | R/W | Enhanced Acoustics Reg. 1 | MIN3 | MIN2 | MIN1 | SYNC | EN1 | ACOU | ACOU | ACOU | 0x00 | Yes |
| 0x63 | R/W | Enhanced Acoustics Reg. 2 | EN2 | ACOU2 | ACOU2 | ACOU2 | EN3 | ACOU3 | ACOU3 | ACOU3 | 0x00 | Yes |
| 0x64 | R/W | PWM1 Min Duty Cycle | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x80 | Yes |
| 0x65 | R/W | PWM2 Min Duty Cycle | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x80 | Yes |
| 0x66 | R/W | PWM3 Min Duty Cycle | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x80 | Yes |
| 0x67 | R/W | Remote 1 Temp. $T_{MIN}$ | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x9A | Yes |
| 0x68 | R/W | Local Temp. $T_{MIN}$ | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x9A | Yes |
| 0x69 | R/W | Remote 2 Temp. $T_{MIN}$ | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x9A | Yes |
| 0x6A | R/W | Remote 1 $\overline{\text{THERM}}$ Temp. Limit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0xA4 | Yes |
| 0x6B | R/W | Local $\overline{\text{THERM}}$ Temp. Limit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0xA4 | Yes |
| 0x6C | R/W | Remote 2 $\overline{\text{THERM}}$ Temp. Limit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0xA4/0 xC8 | Yes |
| 0x6D | R/W | Remote 1 and Local Temp/$T_{MIN}$ Hysteresis | HYSR1 | HYSR1 | HYSR1 | HYSR1 | HYSL | HYSL | HYSL | HYSL | 0x44 | Yes |
| 0x6E | R/W | Remote 2 Temp/$T_{MIN}$ Hysteresis | HYSR2 | HYSR2 | HYSR2 | HYRS2 | RES | RES | RES | RES | 0x40 | Yes |
| 0x6F | R/W | XNOR Tree Test Enable | RES | RES | RES | RES | RES | RES | RES | XEN | 0x00 | Yes |

**Table 53. ADT7473/ADT7473−1 REGISTERS** (continued)

| Addr | R/W | Desc | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | De-fault | Lock-able |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0x70 | R/W | Remote 1 Temp. Offset | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x00 | Yes |
| 0x71 | R/W | Local Temp. Offset | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x00 | Yes |
| 0x72 | R/W | Remote 2 Temp. Offset | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0x00 | Yes |
| 0x73 | R/W | Config. Register 2 | SHDN | CONV | ATTN | AVG | FAN3 Detect | FAN2 Detect | FAN1 Detect | FAN Presence DT | 0x00 | Yes |
| 0x74 | R/W | Interrupt Mask Register 1 | OOL | R2T | LT | R1T | RES | $V_{CC}$ | $V_{CCP}$ | RES | 0x00 | − |
| 0x75 | R/W | Interrupt Mask Register 2 | D2 | D1 | F4P | FAN3 | FAN2 | FAN1 | OVT | RES | 0x00 | − |
| 0x76 | R | Extended Resolution 1 | RES | RES | $V_{CC}$ | $V_{CC}$ | $V_{CCP}$ | $V_{CCP}$ | RES | RES | 0x00 | − |
| 0x77 | R | Extended Resolution 2 | TDM2 | TDM2 | LTMP | LTMP | TDM1 | TDM1 | RES | RES | 0x00 | − |
| 0x78 | R/W | Config. Register 3 | DC4 | DC3 | DC2 | DC1 | FAST | BOOST | $\overline{\text{THERM}}$ | ALERT Enable | 0x00 | Yes |
| 0x79 | R | $\overline{\text{THERM}}$ Timer Status Register | TMR | TMR | TMR | TMR | TMR | TMR | TMR | ASRT/ TMRO | 0x00 | − |
| 0x7A | R/W | $\overline{\text{THERM}}$ Timer Limit Register | LIMT | LIMT | LIMT | LIMT | LIMT | LIMT | LIMT | LIMT | 0x00 | − |
| 0x7B | R/W | TACH Pulses per Revolution | FAN4 | FAN4 | FAN3 | FAN3 | FAN2 | FAN2 | FAN1 | FAN1 | 0X55 | − |
| 0x7C | R/W | Config. Register 5 | R2 $\overline{\text{THERM}}$ | Local $\overline{\text{THERM}}$ | R1 $\overline{\text{THERM}}$ | ADT7473: RES ADT7473−1: $\overline{\text{THERM}}$ HYSTR | GPIOP | GPIOD | Temp Offset | TWOS COMPL | ADT 7473: 0x00 | Yes |
| 0x7D | R/W | Config. Register 4 | RES | RES | BpAtt $V_{CCP}$ | ADT7473: RES ADT7473−1: $\overline{\text{THERM}}$_LATCH CONFIG | Max/ Full on $\overline{\text{THERM}}$ | $\overline{\text{THERM}}$ Disable | PIN9 FUNC | PIN9 FUNC | 0x00 | Yes |
| 0x7E | R | Test Register 1 | DO NOT WRITE TO THESE REGISTERS | | | | | | | | 0x00 | Yes |
| 0x7F | R | Test Register 2 | DO NOT WRITE TO THESE REGISTERS | | | | | | | | 0x00 | Yes |
| 0x80 | R | Test Register 3 | DO NOT WRITE TO THESE REGISTERS | | | | | | | | 0x10 | Yes |

**Table 54. VOLTAGE READING REGISTERS (POWER-ON DEFAULT = 0X00)** (Note 1)

| Register Address | R/W | Description |
|---|---|---|
| 0x21 | Read-only | Reflects the voltage measurement at the $V_{CCP}$ input on Pin 14 (8 MSB of reading). (Note 2) |
| 0x22 | Read-only | Reflects the voltage measurement at the $V_{CC}$ input on Pin 3 (8 MSB of reading). (Note 3) |

1. If the extended resolution bits of these readings are also being read, the extended resolution registers (Register 0x76 and Register 0x77) must be read first. Once the extended resolution registers have been read, the associated MSB reading registers are frozen until read. Both the extended resolution registers and the MSB registers are frozen.
2. If $V_{CCP}$LO (Bit 1 of the Dynamic $T_{MIN}$ Control Register 1, 0x36) is set, $V_{CCP}$ can control the sleep state of the ADT7473/ADT7473−1.
3. $V_{CC}$ (Pin 3) is the supply voltage for the ADT7473/ADT7473−1.

**Table 55. TEMPERATURE READING REGISTERS (POWER-ON DEFAULT = 0X01)** (Note 1 and 2)

| Register Address | R/W | Description |
|---|---|---|
| 0x25 | Read-only | Remote 1 Temperature Reading (8 MSB of Reading) (Note 3 and 4) |
| 0x26 | Read-only | Local Temperature Reading (8 MSB of Reading) |
| 0x27 | Read-only | Remote 2 Temperature Reading (8 MSB of Reading) (Note 3 and 4) |

1. These temperature readings can be in twos complement or Offset 64 format; this interpretation is determined by Bit 0 of Configuration Register 5 (0x7C).
2. If the extended resolution bits of these readings are also being read, the extended resolution registers (Register 0x76 and Register 0x77) must be read first. Once the extended resolution registers have been read, all associated MSB reading registers are frozen until read. Both the extended resolution registers and the MSB registers are frozen.
3. In twos complement mode, a temperature reading of −128°C (0x80) indicates a diode fault (open or short) on that channel.
4. In Offset 64 mode, a temperature reading of −64°C (0x00) indicates a diode fault (open or short) on that channel.

**Table 56. FAN TACHOMETER READING REGISTERS (POWER-ON DEFAULT = 0X00)** (Note 1)

| Register Address | R/W | Description |
|---|---|---|
| 0x28 | Read-only | TACH1 Low Byte |
| 0x29 | Read-only | TACH1 High Byte |
| 0x2A | Read-only | TACH2 Low Byte |
| 0x2B | Read-only | TACH2 High Byte |
| 0x2C | Read-only | TACH3 Low Byte |
| 0x2D | Read-only | TACH3 High Byte |
| 0x2E | Read-only | TACH4 Low Byte |
| 0x2F | Read-only | TACH4 High Byte |

1. These registers count the number of 11.11 μs periods (based on an internal 90 kHz clock) that occur between a number of consecutive fan TACH pulses (default = 2). The number of TACH pulses used to count can be changed using the TACH pulses per revolution register (Register 0x7B). This allows the fan speed to be accurately measured. Because a valid fan tachometer reading requires that two bytes are read, the low byte must be read first. Both the low and high bytes are then frozen until read. At power-on, these registers contain 0x0000 until the first valid fan TACH measurement is read into these registers. This prevents false interrupts from occurring while the fans are spinning up. A count of 0xFFFF indicates a fan is one of the following:
   - Stalled or blocked (object jamming the fan).
   - Failed (internal circuitry destroyed).
   - Not populated. (The ADT7473/ADT7473−1 expects to see a fan connected to each TACH. If a fan is not connected to that TACH, its TACH minimum high and low bytes should be set to 0xFFFF.)
   - Alternate function, for example, TACH4 reconfigured as $\overline{\text{THERM}}$ pin.

**Table 57. CURRENT PWM DUTY CYCLE REGISTERS (ADT7473 POWER-ON DEFAULT = 0X00, ADT7473−1 POWER-ON DEFAULT = 0XFF)** (Note 1)

| Register Address | R/W | Description |
|---|---|---|
| 0x30 | R/W | PWM1 Current Duty Cycle (0% to 100% Duty Cycle = 0x00 to 0xFF) |
| 0x31 | R/W | PWM2 Current Duty Cycle (0% to 100% Duty Cycle = 0x00 to 0xFF) |
| 0x32 | R/W | PWM3 Current Duty Cycle (0% to 100% Duty Cycle = 0x00 to 0xFF) |

1. These registers reflect the PWM duty cycle driving each fan at any given time. When in automatic fan speed control mode, the ADT7473/ADT7473−1 reports the PWM duty cycles back through these registers. The PWM duty cycle values vary according to temperature in automatic fan speed control mode. During fan startup, these registers report back 0x00. In software mode, the PWM duty cycle outputs can be set to any duty cycle value by writing to these registers.

**Table 58. OPERATING POINT REGISTERS (POWER-ON = 0XA4)** (Note 1, 2 and 3)

| Register Address | **R/W** (Note 3) | Description |
|---|---|---|
| 0x33 | R/W | Remote 1 Operating Point Register (Default = 100°C) |
| 0x34 | R/W | Local Temperature Operating Point Register (Default = 100°C) |
| 0x35 | R/W | Remote 2 Operating Point Register (Default = 100°C) |

1. These registers set the target operating point for each temperature channel when the dynamic $T_{MIN}$ control feature is enabled.
2. The fans being controlled are adjusted to maintain temperature about an operating point.
3. These registers become read-only when the Configuration Register 1 Lock bit is set to 1. Any subsequent attempts to write to these registers fail.

**Table 59. REGISTER 0X36 − DYNAMIC $T_{MIN}$ CONTROL REGISTER 1 (POWER-ON DEFAULT = 0X00)** (Note 1)

| Bit No. | Mnemonic | R/W | Description |
|---|---|---|---|
| <0> | CYR2 | R/W | MSB of 3-bit Remote 2 cycle value. The other two bits of the code reside in Dynamic $T_{MIN}$ Control Register 2 (Reg. 0x37). These three bits define the delay time between making subsequent $T_{MIN}$ adjustments in the control loop, in terms of the number of monitoring cycles. The system has associated thermal time constants that need to be found to optimize the response of fans and the control loop. |
| <1> | $V_{CCP}$LO | R/W | $V_{CCP}$LO = 1. When the power is supplied from 3.3 V STANDBY and the core voltage ($V_{CCP}$) drops below its $V_{CCP}$ low limit value (Register 0x46), the following occurs:<br>Status Bit 1 in Interrupt Status Register 1 is set.<br>$\overline{\text{SMBALERT}}$ is generated, if enabled.<br>$\overline{\text{PROCHOT}}$ monitoring is disabled.<br>Dynamic $T_{MIN}$ control is disabled.<br>The device is prevented from entering shutdown.<br>Everything is re-enabled once $V_{CCP}$ increases above the $V_{CCP}$ low limit. |
| <2> | PHTR1 | R/W | PHTR1 = 1 copies the Remote 1 current temperature to the Remote 1 operating point register if $\overline{\text{THERM}}$ is asserted. The operating point contains the temperature at which $\overline{\text{THERM}}$ is asserted, allowing the system to run as quietly as possible without affecting system performance.<br>PHTR1 = 0 ignores any $\overline{\text{THERM}}$ assertions on the $\overline{\text{THERM}}$ pin. The Remote 1 operating point register reflects its programmed value. |
| <3> | PHTL | R/W | PHTL = 1 copies the local channel's current temperature to the local operating point register if $\overline{\text{THERM}}$ is asserted. The operating point contains the temperature at which $\overline{\text{THERM}}$ is asserted. This allows the system to run as quietly as possible without affecting system performance.<br>PHTL = 0 ignores any $\overline{\text{THERM}}$ assertions on the $\overline{\text{THERM}}$ pin. The local temperature operating point register reflects its programmed value. |
| <4> | PHTR2 | R/W | PHTR2 = 1 copies the Remote 2 current temperature to the Remote 2 operating point register if $\overline{\text{THERM}}$ is asserted. The operating point contains the temperature at which $\overline{\text{THERM}}$ is asserted, allowing the system to run as quietly as possible without affecting system performance.<br>PHTR2 = 0 ignores any $\overline{\text{THERM}}$ assertions on the $\overline{\text{THERM}}$ pin. The Remote 2 operating point register reflects its programmed value. |
| <5> | R1T | R/W | R1T = 1 enables dynamic $T_{MIN}$ control on the Remote 1 temperature channel. The chosen $T_{MIN}$ value is dynamically adjusted based on the current temperature, operating point, and high and low limits for this zone.<br>R1T = 0 disables dynamic $T_{MIN}$ control. The $T_{MIN}$ value chosen is not adjusted, and the channel behaves as described in the Fan Speed Control section. |
| <6> | LT | R/W | LT = 1 enables dynamic $T_{MIN}$ control on the local temperature channel. The chosen $T_{MIN}$ value is dynamically adjusted based on the current temperature, operating point, and high and low limits for this zone.<br>LT = 0 disables dynamic $T_{MIN}$ control. The $T_{MIN}$ value chosen is not adjusted, and the channel behaves as described in the Fan Speed Control section. |
| <7> | R2T | R/W | R2T = 1 enables dynamic $T_{MIN}$ control on the Remote 2 temperature channel. The chosen $T_{MIN}$ value is dynamically adjusted based on the current temperature, operating point, and high and low limits for this zone.<br>R2T = 0 disables dynamic $T_{MIN}$ control. The $T_{MIN}$ value chosen is not adjusted and the channel behaves as described in the Fan Speed Control section. |

1. This register becomes read-only when the Configuration Register 1 Lock bit is set to 1. Any subsequent attempts to write to this register fail.

**Table 60. REGISTER 0X37 − DYNAMIC $T_{MIN}$ CONTROL REGISTER 2 (POWER-ON DEFAULT = 0X00)** (Note 1)

| Bit No. | Mnemonic | R/W | Description |
|---|---|---|---|
| <2:0> | CYR1 | R/W | 3-bit Remote 1 cycle value. These three bits define the delay time between making subsequent $T_{MIN}$ adjustments in the control loop for the Remote 1 channel, in terms of number of monitoring cycles. The system has associated thermal time constants that need to be found to optimize the response of fans and the control loop.<br>**Bits** **Decrease Cycle** **Increase Cycle**<br>000 8 cycles (1 sec) 16 cycles (2 sec)<br>001 16 cycles (2 sec) 32 cycles (4 sec)<br>010 32 cycles (4 sec) 64 cycles (8 sec)<br>011 64 cycles (8 sec) 128 cycles (16 sec)<br>100 128 cycles (16 sec) 256 cycles (32 sec)<br>101 256 cycles (32 sec) 512 cycles (64 sec)<br>110 512 cycles (64 sec) 1024 cycles (128 sec)<br>111 1024 cycles (128 sec) 2048 cycles (256 sec) |
| <5:3> | CYL | R/W | 3-bit Local temperature cycle value. These three bits define the delay time between making subsequent $T_{MIN}$ adjustments in the control loop for the local temperature channel, in terms of number of monitoring cycles. The system has associated thermal time constants that need to be found to optimize the response of fans and the control loop.<br>**Bits** **Decrease Cycle** **Increase Cycle**<br>000 8 cycles (1 sec) 16 cycles (2 sec)<br>001 16 cycles (2 sec) 32 cycles (4 sec)<br>010 32 cycles (4 sec) 64 cycles (8 sec)<br>011 64 cycles (8 sec) 128 cycles (16 sec)<br>100 128 cycles (16 sec) 256 cycles (32 sec)<br>101 256 cycles (32 sec) 512 cycles (64 sec)<br>110 512 cycles (64 sec) 1024 cycles (128 sec)<br>111 1024 cycles (128 sec) 2048 cycles (256 sec) |
| <7:6> | CYR2 | R/W | 2 LSBs of 3-bit Remote 2 cycle value. The MSB of the 3-bit code resides in Dynamic $T_{MIN}$ Control Register 1 (Register 0x36). These three bits define the delay time between making subsequent $T_{MIN}$ adjustments in the control loop for the Remote 2 channel, in terms of number of monitoring cycles. The system has associated thermal time constants that need to be found to optimize the response of fans and the control loop.<br>**Bits** **Decrease Cycle** **Increase Cycle**<br>000 8 cycles (1 sec) 16 cycles (2 sec)<br>001 16 cycles (2 sec) 32 cycles (4 sec)<br>010 32 cycles (4 sec) 64 cycles (8 sec)<br>011 64 cycles (8 sec) 128 cycles (16 sec)<br>100 128 cycles (16 sec) 256 cycles (32 sec)<br>101 256 cycles (32 sec) 512 cycles (64 sec)<br>110 512 cycles (64 sec) 1024 cycles (128 sec)<br>111 1024 cycles (128 sec) 2048 cycles (256 sec) |

1. This register becomes read-only when the Configuration Register 1 Lock bit is set to 1. Any subsequent attempts to write to this register fail.

**Table 61. MAXIMUM PWM DUTY CYCLE REGISTERS (POWER-ON DEFAULT = 0XFF)** (Note 1, 2, and 3)

| Register Address | **R/W** (Note 2) | Description |
|---|---|---|
| 0x38 | R/W | Maximum Duty Cycle for PWM1 Output, Default = 100% (0xFF) |
| 0x39 | R/W | Maximum Duty Cycle for PWM2 Output, Default = 100% (0xFF) |
| 0x3A | R/W | Maximum Duty Cycle for PWM3 Output, Default = 100% (0xFF) |

1. These registers set the maximum PWM duty cycle of the PWM output.
2. These registers become read-only when the Configuration Register 1 Lock bit is set to 1. Any subsequent attempts to write to this register fail.
3. If Bit 3 of Configuration Register 4 (0x7D) is set, then on a $\overline{\text{THERM}}$ overtemperature event, fans go to their maximum programmed PWM value as programmed here. If Bit 3 of Configuration Register 4 (0x7D) is 0, then on a $\overline{\text{THERM}}$ overtemperature event, fans go to 100% PWM.

**Table 62. REGISTER 0X40 − CONFIGURATION REGISTER 1 (POWER-ON DEFAULT = 0X01)**

| Bit No. | Mnemonic | R/W | Description |
|---|---|---|---|
| <0> | STRT | R/W | Logic 1 enables monitoring and PWM control outputs based on the limit settings programmed. Logic 0 disables monitoring and PWM control based on the default powerup limit settings. This bit is not locked when Bit 1 (LOCK bit) has been written. This bit remains writable after lock bit is set. |
| <1> | LOCK | Write Once | Logic 1 locks all limit values to their current settings. Once this bit is set, all lockable registers become read-only and cannot be modified until the ADT7473/ADT7473−1 is powered down and powered up again. This prevents rogue programs, such as viruses, from modifying critical system limit settings. (This is a lockable bit.) |
| <2> | RDY | Read-only | This bit is set to 1 by the ADT7473/ADT7473−1 to indicate only that the device is fully powered up and ready to begin system monitoring. |
| <3> | FSPD | R/W | When set to 1, this bit runs all fans at maximum speed as programmed in the maximum PWM duty cycle registers (0x30, 0x38, 0x39 and 0x3A ). Power-on default = 0. This bit is not locked at any time. |
| <4> | Vx1 | R/W | BIOS should set this bit to a 1 when the ADT7473/ADT7473−1 is configured to measure current from an ADI ADOPT™ VRM controller and to measure the CPU's core voltage. This bit allows monitoring software to display CPU watts usage. (This is a lockable bit.) |
| <5> | FSPDIS | R/W | Logic 1 disables fan spin-up for two TACH pulses. Instead, the PWM outputs go high for the entire fan spin-up timeout selected. |
| <6> | TODIS | R/W | When this bit is set to 1, the SMBus timeout feature is enabled. This allows the ADT7473/ADT7473−1 to be used with SMBus controllers that cannot handle SMBus timeouts. (This is a lockable bit.) |
| <7> | RES<br>Latch Reset | − | Reserved on the ADT7473.<br>On the ADT7473−1, resets latch conditions when set to 1. |

**Table 63. REGISTER 0X41 − INTERRUPT STATUS REGISTER 1 (POWER-ON DEFAULT = 0X00)**

| Bit No. | Mnemonic | R/W | Description |
|---|---|---|---|
| <1> | $V_{CCP}$ | Read-only | $V_{CCP}$ = 1 indicates the $V_{CCP}$ high or low limit has been exceeded. This bit is cleared on a read of the status register only if the error condition has subsided. |
| <2> | $V_{CC}$ | Read-only | $V_{CC}$ = 1 indicates the $V_{CC}$ high or low limit has been exceeded. This bit is cleared on a read of the status register only if the error condition has subsided. |
| <4> | R1T | Read-only | R1T = 1 indicates the Remote 1 low or high temperature has been exceeded. This bit is cleared on a read of the status register only if the error condition has subsided. |
| <5> | LT | Read-only | LT = 1 indicates the local low or high temperature has been exceeded. This bit is cleared on a read of the status register only if the error condition has subsided. |
| <6> | R2T | Read-only | R2T = 1 indicates the Remote 2 low or high temperature has been exceeded. This bit is cleared on a read of the status register only if the error condition has subsided. |
| <7> | OOL | Read-only | OOL = 1 indicates an out-of-limit event has been latched in Interrupt Status Register 2 (0x42). This bit is a logical OR of all status bits in Interrupt Status Register 2. Software can test this bit in isolation to determine whether any of the voltage, temperature, or fan speed readings represented by Interrupt Status Register 2 are out-of-limit, which saves the need to read Interrupt Status Register 2 every interrupt or polling cycle. |

**Table 64. REGISTER 0X42 − INTERRUPT STATUS REGISTER 2 (POWER-ON DEFAULT = 0X00)**

| Bit No. | Mnemonic | R/W | Description |
|---|---|---|---|
| <0> | RES<br>$\overline{\text{THERM}}$ Limit Latch | Read-only | Reserved on the ADT7473<br>On the ADT7473−1, $\overline{\text{THERM}}$ Limit Latch = 1 indicates Remote Channel 2 latch. This is a $\overline{\text{THERM}}$ limit condition. |
| <1> | OVT | Read-only | OVT = 1 indicates one of the $\overline{\text{THERM}}$ overtemperature limits is exceeded. This bit is cleared on a read of the status register when the temperature drops below $\overline{\text{THERM}}$ – $T_{HYST}$. |
| <2> | FAN1 | Read-only | FAN1 = 1 indicates Fan 1 has dropped below minimum speed or has stalled. This bit is not set when the PWM1 output is off. |
| <3> | FAN2 | Read-only | FAN2 = 1 indicates Fan 2 has dropped below minimum speed or has stalled. This bit is not set when the PWM2 output is off. |
| <4> | FAN3 | Read-only | FAN3 = 1 indicates Fan 3 has dropped below minimum speed or has stalled. This bit is not set when the PWM3 output is off. |
| <5> | F4P | Read-only<br>R/W<br>Read-only | F4P = 1 indicates Fan 4 has dropped below minimum speed or has stalled. This bit is not set when the PWM3 output is off.<br>When Pin 9 is programmed as a GPIO output, writing to this bit determines the logic output of the GPIO.<br>If Pin 9 is configured as the $\overline{\text{THERM}}$ timer input for $\overline{\text{THERM}}$ monitoring, then this bit is set when the $\overline{\text{THERM}}$ assertion time exceeds the limit programmed in the $\overline{\text{THERM}}$ timer limit register (Reg. 0x7A). |
| <6> | D1 | Read-only | D1 = 1 indicates either an open or short circuit on the Thermal Diode 1 inputs. |
| <7> | D2 | Read-only | D2 = 1 indicates either an open or short circuit on the Thermal Diode 2 inputs. |

**Table 65. VOLTAGE LIMIT REGISTERS** (Note 1)

| Register Address | R/W | Description (Note 2) | Power-On Default |
|---|---|---|---|
| 0x46 | R/W | $V_{CCP}$ Low Limit | 0x00 |
| 0x47 | R/W | $V_{CCP}$ High Limit | 0xFF |
| 0x48 | R/W | $V_{CC}$ Low Limit | 0x00 |
| 0x49 | R/W | $V_{CC}$ High Limit | 0xFF |

1. Setting the Configuration Register 1 Lock bit has no effect on these registers.
2. High limits: an interrupt is generated when a value exceeds its high limit (> comparison). Low limits: an interrupt is generated when a value is equal to or below its low limit (≤ comparison).

**Table 66. TEMPERATURE LIMIT REGISTERS** (Note 1)

| Register Address | R/W | Description (Note 2) | Power-On Default |
|---|---|---|---|
| 0x4E | R/W | Remote 1 Temperature Low Limit | 0x01 |
| 0x4F | R/W | Remote 1 Temperature High Limit | 0xFF |
| 0x50 | R/W | Local Temperature Low Limit | 0x01 |
| 0x51 | R/W | Local Temperature High Limit | 0xFF |
| 0x52 | R/W | Remote 2 Temperature Low Limit | 0x01 |
| 0x53 | R/W | Remote 2 Temperature High Limit | 0xFF |

1. Exceeding any of these temperature limits by 1°C causes the appropriate status bit to be set in the interrupt status register. Setting the Configuration Register 1 Lock bit has no effect on these registers.
2. High limits: an interrupt is generated when a value exceeds its high limit (> comparison). Low limits: an interrupt is generated when a value is equal to or below its low limit (≤ comparison).

**Table 67. FAN TACHOMETER LIMIT REGISTERS** (Note 1)

| Register Address | R/W | Description | Power-On Default |
|---|---|---|---|
| 0x54 | R/W | TACH1 Minimum Low Byte | 0xFF |
| 0x55 | R/W | TACH1 Minimum High Byte/Single-channel ADC Channel Select | 0xFF |
| 0x56 | R/W | TACH2 Minimum Low Byte | 0xFF |
| 0x57 | R/W | TACH2 Minimum High Byte | 0xFF |
| 0x58 | R/W | TACH3 Minimum Low Byte | 0xFF |
| 0x59 | R/W | TACH3 Minimum High Byte | 0xFF |
| 0x5A | R/W | TACH4 Minimum Low Byte | 0xFF |
| 0x5B | R/W | TACH4 Minimum High Byte | 0xFF |

1. Exceeding any of the TACH limit registers by 1 indicates that the fan is running slowly or has stalled. The appropriate status bit is set in Interrupt Status Register 2 to indicate the fan failure. Setting the Configuration Register 1 Lock bit has no effect on these registers.

**Table 68. REGISTER 0X55 − TACH1 MINIMUM HIGH BYTE (POWER-ON DEFAULT = 0XFF)**

| Bit No. | Mnemonic | R/W | Description |
|---|---|---|---|
| <4:0> | Reserved | Read-only | These bits are reserved when Bit 6 of Configuration Register 2 (0x73) is set (single-channel ADC mode). Otherwise, these bits represent Bits <4:0> of the TACH1 minimum high byte. |
| <7:5> | SCADC | R/W | When Bit 6 of Configuration Register 2 (0x73) is set (single-channel ADC mode), these bits are used to select the only channel from which the ADC makes measurements. Otherwise, these bits represent Bits <7:5> of the TACH1 minimum high byte. |

**Table 69. PWM CONFIGURATION REGISTERS**

| Register Address | R/W (Note 1) | Description | Power-On Default |
|---|---|---|---|
| 0x5C | R/W | PWM1 Configuration | ADT7473: 0x82<br>ADT7473−1: 0x62 |
| 0x5D | R/W | PWM2 Configuration | ADT7473: 0x82<br>ADT7473−1: 0x62 |
| 0x5E | R/W | PWM3 Configuration | ADT7473: 0x82<br>ADT7473−1: 0x62 |

1. These registers become read-only when the Configuration Register 1 Lock bit is set. Any further attempts to write to these registers have no effect.

**Table 70. REGISTER 0X5C, REGISTER 0X5D, AND REGISTER 0X5E − CONFIGURATION REGISTERS (ADT7473 POWER-ON DEFAULT = 0X82, ADT7473−1 POWER-ON DEFAULT = 0X62)**

| Bit No. | Mnemonic | R/W | Description |
|---|---|---|---|
| <2:0> | SPIN | R/W | These bits control the startup timeout for PWMx. The PWM output stays high until two valid TACH rising edges are seen from the fan. If there is not a valid TACH signal during the fan TACH measurement directly after the fan startup timeout period, then the TACH measurement reads 0xFFFF and Interrupt Status Register 2 reflects the fan fault. If the TACH minimum high and low bytes contain 0xFFFF or 0x0000, then the Interrupt Status Register 2 bit is not set, even if the fan has not started.<br>**Bit Code** **Startup Time**<br>000 No Startup Timeout<br>001 100 ms<br>010 250 ms (Default)<br>011 400 ms<br>100 667 ms<br>101 1 sec<br>110 2 sec<br>111 4 sec |
| <3> | SLOW | R/W | SLOW = 1 makes the ramp rates for acoustic enhancement four times longer. |
| <4> | INV | R/W | This bit inverts the PWM output. The default is 0, which corresponds to a logic high output for 100% duty cycle. Setting this bit to 1 inverts the PWM output, so 100% duty cycle corresponds to a logic low output. |
| <7:5> | BHVR | R/W | These bits assign each fan to a particular temperature sensor for localized cooling.<br>000 = Remote 1 temperature controls PWMx (automatic fan control mode).<br>001 = Local temperature controls PWMx (automatic fan control mode).<br>010 = Remote 2 temperature controls PWMx (automatic fan control mode).<br>011 = PWMx runs full speed (default for ADT7473−1).<br>100 = PWMx disabled (default for ADT7473).<br>101 = Fastest speed calculated by local and Remote 2 temperature controls PWMx.<br>110 = Fastest speed calculated by all three temperature channel controls PWMx.<br>111 = Manual mode. PWM duty cycle registers (Register 0x30 to Register 0x32) become writable. |

**Table 71. TEMPERATURE $T_{RANGE}$/PWM FREQUENCY REGISTERS**

| Register Address | R/W (Note 1) | Description | Power-On Default |
|---|---|---|---|
| 0x5F | R/W | Remote 1 $T_{RANGE}$/PWM1 Frequency | 0xCC |
| 0x60 | R/W | Local Temperature $T_{RANGE}$/PWM2 Frequency | 0xCC |
| 0x61 | R/W | Remote 2 $T_{RANGE}$/PWM3 Frequency | 0xCC |

1. These registers become read-only when the Configuration Register 1 Lock bit is set. Any further attempts to write to these registers have no effect.

**Table 72. REGISTER 0X5F, REGISTER 0X60, AND REGISTER 0X61 − TEMP $T_{RANGE}$/PWM FREQUENCY REGISTERS, (POWER-ON DEFAULT = 0XCC)**

| Bit No. | Mnemonic | R/W | Description |
|---|---|---|---|
| <2:0> | FREQ | R/W | These bits control the PWMx frequency.<br>**Bit Code** **Frequency**<br>000 11.0 Hz<br>001 14.7 Hz<br>010 22.1 Hz<br>011 29.4 Hz<br>100 35.3 Hz (Default)<br>101 44.1 Hz<br>110 58.8 Hz<br>111 88.2 Hz |
| <3> | HF/LF | R/W | HF/LF = 1, high frequency PWM output for 4-wire fans. Once enabled, 3-wire fan-specific settings have no effect.<br>0x5F, HF/LF = 1, Enables High Frequency Mode for Fan 1<br>0x60, HF/LF = 1, Enables High Frequency Mode for Fan 2<br>0x61, HF/LF = 1, Enables High Frequency Mode for Fan 3 |
| <7:4> | RANGE | R/W | These bits determine the PWM duty cycle vs. the temperature slope for automatic fan control.<br>**Bit Code** **Temperature**<br>0000 2°C<br>0001 2.5°C<br>0010 3.33°C<br>0011 4°C<br>0100 5°C<br>0101 6.67°C<br>0110 8°C<br>0111 10°C<br>1000 13.33°C<br>1001 16°C<br>1010 20°C<br>1011 26.67°C<br>1100 32°C (Default)<br>1101 40°C<br>1110 53.33°C<br>1111 80°C |

**Table 73. REGISTER 0X62 − ENHANCED ACOUSTICS REGISTER 1 (POWER-ON DEFAULT = 0X00)**

| Bit No. | Mnemonic | R/W (Note 1) | Description |
|---|---|---|---|
| <2:0> | ACOU | R/W | These bits select the ramp rate applied to the PWM1 output. Instead of PWM1 jumping instantaneously to its newly calculated speed, PWM1 ramps gradually at the rate determined by these bits. This feature enhances the acoustics of the fan being driven by the PWM1 output.<br>**Time Slot Increase** — **Time for 33% to 100%**<br>000 = 1 — 35 sec<br>001 = 2 — 17.6 sec<br>010 = 3 — 11.8 sec<br>011 = 4 — 7.0 sec<br>100 = 8 — 4.4 sec<br>101 = 12 — 3.0 sec<br>110 = 24 — 1.6 sec<br>111 = 48 — 0.8 sec |
| <3> | EN1 | R/W | When this bit is 1, acoustic enhancement is enabled on PWM1 output. |
| <4> | SYNC | R/W | SYNC = 1 synchronizes fan speed measurements on TACH2, TACH3, and TACH4 to PWM3. This allows up to three fans to be driven from PWM3 output and their speeds to be measured. SYNC = 0 synchronizes only TACH3 and TACH4 to PWM3 output. |
| <5> | MIN1 | R/W | When the ADT7473/ADT7473−1 is in automatic fan control mode, this bit defines whether PWM1 is off (0% duty cycle) or at PWM1 minimum duty cycle when the controlling temperature is below its $T_{MIN}$ – hysteresis value.<br>0 = 0% duty cycle below $T_{MIN}$ − hysteresis.<br>1 = PWM1 minimum duty cycle below $T_{MIN}$ − hysteresis. |
| <6> | MIN2 | R/W | When the ADT7473/ADT7473−1 is in automatic fan speed control mode, this bit defines whether PWM2 is off (0% duty cycle) or at PWM2 minimum duty cycle when the controlling temperature is below its $T_{MIN}$ − hysteresis value.<br>0 = 0% duty cycle below $T_{MIN}$ − hysteresis.<br>1 = PWM 2 minimum duty cycle below $T_{MIN}$ − hysteresis. |
| <7> | MIN3 | R/W | When the ADT7473/ADT7473−1 is in automatic fan speed control mode, this bit defines whether PWM3 is off (0% duty cycle) or at PWM3 minimum duty cycle when the controlling temperature is below its $T_{MIN}$ − hysteresis value.<br>0 = 0% duty cycle below $T_{MIN}$ − hysteresis.<br>1 = PWM3 minimum duty cycle below $T_{MIN}$ − hysteresis. |

1. This register becomes read-only when the Configuration Register 1 Lock bit is set to 1. Any further attempts to write to this register have no effect.

**Table 74. Register 0x63 − Enhanced Acoustics Register 2 (Power-On Default = 0x00)**

| Bit No. | Mnemonic | R/W (Note 1) | Description |
|---|---|---|---|
| <2:0> | ACOU3 | R/W | These bits select the ramp rate applied to the PWM3 output. Instead of PWM3 jumping instantly to its newly calculated speed, PWM3 ramps gradually at the rate determined by these bits. This effect enhances the acoustics of the fan being driven by the PWM3 output.<br>**Time Slot Increase** — **Time for 33% to 100%**<br>000 = 1 — 35 sec<br>001 = 2 — 17.6 sec<br>010 = 3 — 11.8 sec<br>011 = 4 — 7.0 sec<br>100 = 8 — 4.4 sec<br>101 = 12 — 3.0 sec<br>110 = 24 — 1.6 sec<br>111 = 48 — 0.8 sec |
| <3> | EN3 | R/W | When this bit is 1, acoustic enhancement is enabled on PWM3 output. |
| <6:4> | ACOU2 | R/W | These bits select the ramp rate applied to the PWM2 output. Instead of PWM2 jumping instantly to its newly calculated speed, PWM2 ramps gradually at the rate determined by these bits. This effect enhances the acoustics of the fans being driven by the PWM2 output.<br>**Time Slot Increase** — **Time for 33% to 100%**<br>000 = 1 — 35 sec<br>001 = 2 — 17.6 sec<br>010 = 3 — 11.8 sec<br>011 = 4 — 7.0 sec<br>100 = 8 — 4.4 sec<br>101 = 12 — 3.0 sec<br>110 = 24 — 1.6 sec<br>111 = 48 — 0.8 sec |
| <7> | EN2 | R/W | When this bit is 1, acoustic enhancement is enabled on PWM2 output. |

1. This register becomes read-only when the Configuration Register 1 Lock bit is set to 1. Any further attempts to write to this register have no effect.

**Table 75. PWM MINIMUM DUTY CYCLE REGISTERS**

| Register Address | R/W (Note 1) | Description | Power-On Default |
|---|---|---|---|
| 0x64 | R/W | PWM1 Minimum Duty Cycle | 0x80 (50% Duty Cycle) |
| 0x65 | R/W | PWM2 Minimum Duty Cycle | 0x80 (50% Duty Cycle) |
| 0x66 | R/W | PWM3 Minimum Duty Cycle | 0x80 (50% Duty Cycle) |

1. These registers become read-only when the Configuration Register 1 Lock bit is set. Any further attempts to write to these registers have no effect.

**Table 76. REGISTER 0X64, REGISTER 0X65, AND REGISTER 0X66 - PWM MINIMUM DUTY CYCLE REGISTERS (POWER-ON DEFAULT = 0X80, 50% DUTY CYCLE)**

| Bit No. | Mnemonic | R/W | Description |
|---|---|---|---|
| <7:0> | PWM Duty Cycle | R/W | These bits define the $PWM_{MIN}$ duty cycle for PWMx.<br>0x00 = 0% Duty Cycle (Fan Off)<br>0x40 = 25% Duty Cycle<br>0x80 = 50% Duty Cycle<br>0xFF = 100% Duty Cycle (Fan Full Speed) |

**Table 77. $T_{MIN}$ REGISTERS** (Note 1)

| Register Address | R/W (Note 2) | Description | Power-On Default |
|---|---|---|---|
| 0x67 | R/W | Remote 1 Temperature $T_{MIN}$ | 0x9A (90°C) |
| 0x68 | R/W | Local Temperature $T_{MIN}$ | 0x9A (90°C) |
| 0x69 | R/W | Remote 2 Temperature $T_{MIN}$ | 0x9A (90°C) |

1. These are the $T_{MIN}$ registers for each temperature channel. When the temperature measured exceeds $T_{MIN}$, the appropriate fan runs at minimum speed and increases with temperature according to $T_{RANGE}$.
2. These registers become read-only when the Configuration Register 1 Lock bit is set. Any further attempts to write to these registers have no effect.

**Table 78. $\overline{\text{THERM}}$ LIMIT REGISTERS** (Note 1)

| Register Address | R/W (Note 2) | Description | Power-On Default |
|---|---|---|---|
| 0x6A | R/W | Remote 1 $\overline{\text{THERM}}$ Limit | 0xA4 (100°C) |
| 0x6B | R/W | Local $\overline{\text{THERM}}$ Limit | 0xA4 (100°C) |
| 0x6C | R/W | Remote 2 $\overline{\text{THERM}}$ Limit | ADT7473: 0xA4 (100°C)<br>ADT7473−1: 0xC8 (136°C) |

1. If any temperature measured exceeds its $\overline{\text{THERM}}$ limit, all PWM outputs drive their fans at 100% duty cycle. This is a fail-safe mechanism incorporated to cool the system in the event of a critical overtemperature. It also ensures some level of cooling in the event that software or hardware locks up. If set to 0x80, this feature is disabled. The PWM output remains at 100% until the temperature drops below $\overline{\text{THERM}}$ Limit − Hysteresis. If the $\overline{\text{THERM}}$ pin is programmed as an output, then exceeding these limits by 0.25°C can cause the $\overline{\text{THERM}}$ pin to assert low as an output.
2. These registers become read-only when the Configuration Register 1 Lock bit is set to 1. Any further attempts to write to these registers have no effect.

**Table 79. TEMPERATURE/$T_{MIN}$ HYSTERESIS REGISTERS** (Note 1)

| Register Address | Bit Name | R/W (Note 2) | Description | Power-On Default |
|---|---|---|---|---|
| 0x6D | | R/W | Remote 1 and local temperature hysteresis. | 0x44 |
| | HYSL <3:0> | | Local temperature hysteresis. 0°C to 15°C of hysteresis can be applied to the local temperature AFC and dynamic $T_{MIN}$ control loops. | |
| | HYSR1 <7:4> | | Remote 1 temperature hysteresis. 0°C to 15°C of hysteresis can be applied to the Remote 1 temperature AFC and dynamic $T_{MIN}$ control loops. | |
| 0x6E | | R/W | Remote 2 temperature hysteresis. | 0x40 |
| | HYSR2 <7:4> | | Local temperature hysteresis. 0°C to 15°C of hysteresis can be applied to the local temperature AFC and dynamic $T_{MIN}$ control loops. | |

1. Each 4 bit value controls the amount of temperature hysteresis applied to a particular temperature channel. Once the temperature for that channel falls below its $T_{MIN}$ value, the fan remains running at $PWM_{MIN}$ duty cycle until the temperature = $T_{MIN}$ – hysteresis. Up to 15°C of hysteresis can be assigned to any temperature channel. The hysteresis value chosen also applies to that temperature channel, if its $\overline{\text{THERM}}$ limit is exceeded. The PWM output being controlled goes to 100%, if the $\overline{\text{THERM}}$ limit is exceeded and remains at 100% until the temperature drops below $\overline{\text{THERM}}$ – hysteresis. For acoustic reasons, it is recommended that the hysteresis value not be programmed less than 4°C. Setting the hysteresis value lower than 4°C causes the fan to switch on and off regularly when the temperature is close to $T_{MIN}$.
2. These registers become read-only when the Configuration Register 1 Lock bit is set to 1. Any further attempts to write to these registers have no effect.

**Table 80. XNOR TREE TEST REGISTER**

| Register Address | Bit Name | R/W (Note 1) | Description | Power-On Default |
|---|---|---|---|---|
| 0x6F | | R/W | XNOR tree test enable register. | 0x00 |
| | XEN <0> | | If the XEN bit is set to 1, the device enters the XNOR tree test mode. Clearing the bit removes the device from the XNOR tree test mode. | |
| | Res <7:1> | | Unused. Do not write to these bits. | |

1. These registers become read-only when the Configuration Register 1 Lock bit is set to 1. Any further attempts to write to these registers have no effect.

**Table 81. REMOTE 1 TEMPERATURE OFFSET REGISTER (0X70)**

| Register Address | R/W (Note 1) | Description | Power-On Default |
|---|---|---|---|
| <7:0> | R/W | Allows a twos complement offset value to be automatically added to or subtracted from the Remote 1 temperature reading. This is to compensate for any inherent system offsets such as PCB trace resistance. LSB value = 0.5°C. | 0x00 |

1. This register becomes read-only when the Configuration Register 1 Lock bit is set to 1. Any further attempts to write to this register have no effect.

**Table 82. LOCAL TEMPERATURE OFFSET REGISTER (0X71)**

| Register Address | R/W (Note 1) | Description | Power-On Default |
|---|---|---|---|
| <7:0> | R/W | Allows a twos complement offset value to be automatically added to or subtracted from the local temperature reading. LSB value = 0.5°C. | 0x00 |

1. This register becomes read-only when the Configuration Register 1 Lock bit is set to 1. Any further attempts to write to this register have no effect.

**Table 83. REMOTE 2 TEMPERATURE OFFSET REGISTER (0X72)**

| Register Address | R/W (Note 1) | Description | Power-On Default |
|---|---|---|---|
| <7:0> | R/W | Allows a twos complement offset value to be automatically added to or subtracted from the Remote 2 temperature reading. This is to compensate for any inherent system offsets such as PCB trace resistance. LSB value = 0.5°C. | 0x00 |

1. This register becomes read-only when the Configuration Register 1 Lock bit is set to 1. Any further attempts to write to this register have no effect.

**Table 84. REGISTER 0X73 − CONFIGURATION REGISTER 2 (POWER-ON DEFAULT = 0X00)**

| Bit No. | Mnemonic | R/W (Note 1) | Description |
|---|---|---|---|
| <0> | FanPresDT | R/W | When FanPresenceDT = 1, the state of Bits <3:1> of Register 0x73 reflects the presence of a 4-wire fan on the appropriate TACH channel. |
| <1> | Fan1Detect | Read-only | Fan1 Detect = 1 indicates a 4-wire fan is connected to the PWM1 input. |
| <2> | Fan2Detect | Read-only | Fan1 Detect = 1 indicates a 4-wire fan is connected to the PWM2 input. |
| <3> | Fan3Detect | Read-only | Fan1 Detect = 1 indicates a 4-wire fan is connected to the PWM3 input. |
| <4> | AVG | R/W | AVG = 1, averaging on the temperature and voltage measurements is turned off. This allows measurements on each channel to be made much faster. |
| <5> | ATTN | R/W | ATTN = 1, the ADT7473/ADT7473−1 removes the attenuators from the $V_{CCP}$ input. The $V_{CCP}$ input can be used for other functions such as connecting up external sensors. |
| <6> | CONV | R/W | CONV = 1, the ADT7473/ADT7473−1 is put into a single-channel ADC conversion mode. In this mode, the ADT7473/ADT7473−1 can be made to read continuously from one input only, for example, Remote 1 temperature. The appropriate ADC channel is selected by writing to Bits <7:5> of TACH1 minimum high byte register (0x55).<br>**Bits <7:5>, Register 0x55**<br>000 Reserved<br>001 $V_{CCP}$<br>010 $V_{CC}$ (3.3 V)<br>011 Reserved<br>100 Reserved<br>101 Remote 1 Temperature<br>110 Local Temperature<br>111 Remote 2 Temperature |
| <7> | Shutdown | R/W | SHDN = 1, ADT7473/ADT7473−1 goes into shutdown mode. All PWM outputs assert low or high, depending on the state of the INV bit, to switch off all fans. |

1. This register becomes read-only when the Configuration Register 1Lock bit is set to 1. Any further attempts to write to this register have no effect.

**Table 85. REGISTER 0X74 − INTERRUPT MASK REGISTER 1 (POWER-ON DEFAULT = 0X00)**

| Bit No. | Mnemonic | R/W | Description |
|---|---|---|---|
| <1> | $V_{CCP}$ | R/W | $V_{CCP}$ = 1, masks $\overline{\text{SMBALERT}}$ for out-of-limit conditions on the $V_{CCP}$ channel. |
| <2> | $V_{CC}$ | R/W | $V_{CC}$ = 1, masks $\overline{\text{SMBALERT}}$ for out-of-limit conditions on the $V_{CC}$ channel. |
| <4> | R1T | R/W | R1T = 1, masks $\overline{\text{SMBALERT}}$ for out-of-limit conditions on the Remote 1 temperature channel. |
| <5> | LT | R/W | LT = 1, masks $\overline{\text{SMBALERT}}$ for out-of-limit conditions on the local temperature channel. |
| <6> | R2T | R/W | R2T = 1, masks $\overline{\text{SMBALERT}}$ for out-of-limit conditions on the Remote 2 temperature channel. |
| <7> | OOL | R/W | OOL = 0, then when one or more alerts are generated in Interrupt Status Register 2, assuming all the mask bits in the Interrupt Mask Register 2 (0x75) = 1, $\overline{\text{SMBALERT}}$ are still asserted.<br>OOL = 1, then when one or more alerts are generated in Interrupt Status Register 2, assuming all the mask bits in the Interrupt Mask Register 2 (0x75) = 1, $\overline{\text{SMBALERT}}$ are not asserted. |

**Table 86. REGISTER 0X75 − INTERRUPT MASK REGISTER 2 (POWER-ON DEFAULT <7:0> = 0X00)**

| Bit No. | Mnemonic | R/W | Description |
|---|---|---|---|
| <1> | OVT | R/W | OVT = 1, masks $\overline{\text{SMBALERT}}$ for overtemperature $\overline{\text{THERM}}$ conditions. |
| <2> | FAN1 | R/W | FAN1 = 1, masks $\overline{\text{SMBALERT}}$ for a Fan 1 fault. |
| <3> | FAN2 | R/W | FAN2 = 1, masks $\overline{\text{SMBALERT}}$ for a Fan 2 fault. |
| <4> | FAN3 | R/W | FAN3 = 1, masks $\overline{\text{SMBALERT}}$ for a Fan 3 fault. |
| <5> | F4P | R/W | F4P = 1, masks $\overline{\text{SMBALERT}}$ for a Fan 4 fault. If the TACH4 pin is being used as the $\overline{\text{THERM}}$ input, this bit masks $\overline{\text{SMBALERT}}$ for a $\overline{\text{THERM}}$ timer event. |
| <6> | D1 | R/W | D1 = 1, masks $\overline{\text{SMBALERT}}$ for a diode open or short on a Remote 1 channel. |
| <7> | D2 | R/W | D2 = 1, masks $\overline{\text{SMBALERT}}$ for a diode open or short on a Remote 2 channel. |

**Table 87. REGISTER 0X76 − EXTENDED RESOLUTION REGISTER 1 (POWER-ON DEFAULT = 0X00)** (Note 1)

| Bit No. | Mnemonic | R/W | Description |
|---|---|---|---|
| <3:2> | $V_{CCP}$ | Read-only | $V_{CCP}$ LSBs. Holds the 2 LSBs of the 10-bit $V_{CCP}$ measurement. |
| <5:4> | $V_{CC}$ | Read-only | $V_{CC}$ LSBs. Holds the 2 LSBs of the 10-bit $V_{CC}$ measurement. |

1. If this register is read, this register and the registers holding the MSB of each reading are frozen until read.

**Table 88. REGISTER 0X77 − EXTENDED RESOLUTION REGISTER 2 (POWER-ON DEFAULT = 0X00)** (Note 1)

| Bit No. | Mnemonic | R/W | Description |
|---|---|---|---|
| <3:2> | TDM1 | Read-only | Remote 1 temperature LSBs. Holds the 2 LSBs of the 10-bit Remote 1 temperature measurement. |
| <5:4> | LTMP | Read-only | Local temperature LSBs. Holds the 2 LSBs of the 10-bit local temperature measurement. |
| <7:6> | TDM2 | Read-only | Remote 2 temperature LSBs. Holds the 2 LSBs of the 10-bit Remote 2 temperature measurement. |

1. If this register is read, this register and the registers holding the MSB of each reading are frozen until read.

**Table 89. REGISTER 0X78 − CONFIGURATION REGISTER 3 (POWER-ON DEFAULT = 0X00)**

| Bit No. | Mnemonic | **R/W** (Note 1) | Description |
|---|---|---|---|
| <0> | ALERT Enable | R/W | ALERT = 0 (default), ADT7473 Pin 5 is configured as PWM2.<br>ALERT = 1, Pin 5 for ADT7473 (PWM2/$\overline{\text{SMBALERT}}$) is configured as an $\overline{\text{SMBALERT}}$ interrupt output to indicate out-of-limit error conditions.<br>ALERT = 0 (default), ADT7473−1 Pin 5 is configured as. $\overline{\text{THERM\_LATCH}}$.<br>ALERT = 1, Pin 5 for ADT7473−1 ($\overline{\text{THERM\_LATCH}}$/PWM2) is configured as PWM2. |
| <1> | $\overline{\text{THERM}}$ | R/W | $\overline{\text{THERM}}$ Enable = 1 enables $\overline{\text{THERM}}$ functionality on Pin 9. Also determined by Bit 0 and Bit 1 (PIN9FUNC) of Configuration Register 4. Direction is controlled by Bit 5, Bit 6, and Bit 7 of Configuration Register 5 (0x7C). When $\overline{\text{THERM}}$ is asserted, if the fans are running and the boost bit is set, the fans run at full speed. $\overline{\text{THERM}}$ can also be programmed so that a timer monitors the duration $\overline{\text{THERM}}$ has been asserted. |
| <2> | BOOST | R/W | When $\overline{\text{THERM}}$ is an input and BOOST = 1, assertion of $\overline{\text{THERM}}$ causes all fans to run at the maximum programmed duty cycle for fail-safe cooling. |
| <3> | FAST | R/W | FAST = 1, enables fast TACH measurements on all channels. This increases the TACH measurement rate from once per second to once every 250 ms (4x). |
| <4> | DC1 | R/W | DC1 = 1, enables TACH measurements to be continuously made on TACH1. Fans must be driven by dc. Setting this bit prevents pulse stretching because it is not required for dc-driven motors. |
| <5> | DC2 | R/W | DC2 = 1, enables TACH measurements to be continuously made on TACH2. Fans must be driven by dc. Setting this bit prevents pulse stretching because it is not required for dc-driven motors. |
| <6> | DC3 | R/W | DC3 = 1, enables TACH measurements to be continuously made on TACH3. Fans must be driven by dc. Setting this bit prevents pulse stretching because it is not required for dc-driven motors. |
| <7> | DC4 | R/W | DC4 = 1, enables TACH measurements to be continuously made on TACH4. Fans must be driven by dc. Setting this bit prevents pulse stretching because it is not required for dc-driven motors. |

1. This register becomes read-only when the Configuration Register 1 Lock bit is set to 1. Any further attempts to write to this register have no effect.

**Table 90. REGISTER 0X79 − $\overline{\text{THERM}}$ TIMER STATUS REGISTER (POWER-ON DEFAULT = 0X00)**

| Bit No. | Mnemonic | R/W | Description |
|---|---|---|---|
| <7:1> | TMR | R | Times how long $\overline{\text{THERM}}$ input is asserted. These seven bits read 0 until the $\overline{\text{THERM}}$ assertion time exceeds 45.52 ms. |
| <0> | ASRT/ TMR0 | R | This bit is set high on the assertion of the $\overline{\text{THERM}}$ input and is cleared on read. If the $\overline{\text{THERM}}$ assertion time exceeds 45.52 ms, this bit is set and becomes the LSB of the 8-bit TMR reading. This allows $\overline{\text{THERM}}$ assertion times from 45.52 ms to 5.82 sec to be reported back with a resolution of 22.76 ms. |

**Table 91. REGISTER 0X7A − $\overline{\text{THERM}}$ TIMER LIMIT REGISTER (POWER-ON DEFAULT = 0X00)**

| Bit No. | Mnemonic | R/W | Description |
|---|---|---|---|
| <7:0> | LIMT | R/W | Sets maximum $\overline{\text{THERM}}$ assertion length allowed before an interrupt is generated. This is an 8-bit limit with a resolution of 22.76 ms allowing $\overline{\text{THERM}}$ assertion limits of 45.52 ms to 5.82 seconds to be programmed. If the $\overline{\text{THERM}}$ assertion time exceeds this limit, Bit 5 (F4P) of Interrupt Status Register 2 (Reg. 0x42) is set. If the limit value is 0x00, an interrupt is generated immediately on the assertion of the $\overline{\text{THERM}}$ input. |

**Table 92. REGISTER 0X7B − TACH PULSES PER REVOLUTION REGISTER (POWER-ON DEFAULT = 0X55)**

| Bit No. | Mnemonic | R/W | Description |
|---|---|---|---|
| <1:0> | FAN1 | R/W | Sets number of pulses to be counted when measuring Fan 1 speed. Can be used to determine fan pulses per revolution for unknown fan type.<br>**Bit Code** **Pulses Counted**<br>00 1<br>01 2 (Default)<br>10 3<br>11 4 |
| <3:2> | FAN2 | R/W | Sets number of pulses to be counted when measuring Fan 2 speed. Can be used to determine fan pulses per revolution for unknown fan type.<br>**Bit Code** **Pulses Counted**<br>00 1<br>01 2 (Default)<br>10 3<br>11 4 |
| <5:4> | FAN3 | R/W | Sets number of pulses to be counted when measuring Fan 3 speed. Can be used to determine fan pulses per revolution for unknown fan type.<br>**Bit Code** **Pulses Counted**<br>00 1<br>01 2 (Default)<br>10 3<br>11 4 |
| <7:6> | FAN4 | R/W | Sets number of pulses to be counted when measuring Fan 4 speed. Can be used to determine fan pulses per revolution for unknown fan type.<br>**Bit Code** **Pulses Counted**<br>00 1<br>01 2 (Default)<br>10 3<br>11 4 |

**Table 93. REGISTER 0X7C − CONFIGURATION REGISTER 5 (ADT7473POWER-ON DEFAULT = 0X00)**

| Bit No. | Mnemonic | R/W (Note 1) | Description |
|---|---|---|---|
| <0> | TWOS COMPL | R/W | Twos complement = 1, sets the temperature range to twos complement temperature range. Twos complement = 0, changes the temperature range to Offset 64. When this bit is changed, the ADT7473/ADT7473−1 interprets all relevant temperature register values as defined by this bit. |
| <1> | Temp Offset | − | TempOffset = 0 sets offset range to ±64°C at 0.5°C resolution.<br>TempOffset = 1 sets offset range to ±128°C at 1°C resolution. |
| <2> | GPIOD | − | GPIO direction. When GPIO function is enabled, this determines whether the GPIO is an input (0) or an output (1). |
| <3> | GPIOP | − | GPIO polarity. When the GPIO function is enabled and is programmed as an output, this bit determines whether the GPIO is active low (0) or high (1). |
| <4> | RES<br>$\overline{\text{THERM}}$ Hysteresis | R/W | Reserved on the ADT7473 On the ADT7473−1:<br>0 = $\overline{\text{THERM}}$ hysteresis disabled<br>1 = $\overline{\text{THERM}}$ hysteresis enabled |
| <5> | R1 $\overline{\text{THERM}}$ | R/W | R1 $\overline{\text{THERM}}$ = 1, $\overline{\text{THERM}}$ temperature limit functionality enabled for Remote 1 temperature channel; that is, $\overline{\text{THERM}}$ is bidirectional. R1 $\overline{\text{THERM}}$ = 0, $\overline{\text{THERM}}$ is a timer input only. $\overline{\text{THERM}}$ can also be disabled on any channel by:<br>Writing −64°C to the appropriate $\overline{\text{THERM}}$ temperature limit in Offset 64 mode.<br>Writing −128°C to the appropriate $\overline{\text{THERM}}$ temperature limit in twos complement mode. |
| <6> | Local $\overline{\text{THERM}}$ | R/W | Local $\overline{\text{THERM}}$ = 1, $\overline{\text{THERM}}$ temperature limit functionality enabled for the local temperature channel; that is, $\overline{\text{THERM}}$ is bidirectional. Local $\overline{\text{THERM}}$ = 0, $\overline{\text{THERM}}$ is a timer input only. $\overline{\text{THERM}}$ can also be disabled on any channel by:<br>Writing −64°C to the appropriate $\overline{\text{THERM}}$ temperature limit in Offset 64 mode.<br>Writing −128°C to the appropriate $\overline{\text{THERM}}$ temperature limit in twos complement mode. |
| <7> | R2 $\overline{\text{THERM}}$ | R/W | R2 $\overline{\text{THERM}}$ = 1, $\overline{\text{THERM}}$ temperature limit functionality enabled for Remote 2 temperature channel; that is, $\overline{\text{THERM}}$ is bidirectional. R2 $\overline{\text{THERM}}$ = 0, $\overline{\text{THERM}}$ is a timer input only. $\overline{\text{THERM}}$ can also be disabled on any channel by:<br>Writing −64°C to the appropriate $\overline{\text{THERM}}$ temperature limit in Offset 64 mode.<br>Writing −128°C to the appropriate $\overline{\text{THERM}}$ temperature limit in twos complement mode. |

1. This register becomes read-only when the Configuration Register 1 Lock bit is set to 1. Any further attempts to write to this register have no effect.

**Table 94. REGISTER 0X7D − CONFIGURATION REGISTER 4 (POWER-ON DEFAULT = 0X00)**

| Bit No. | Mnemonic | R/W (Note 1) | Description |
|---|---|---|---|
| <1:0> | Pin9FUNC | R/W | These bits set the functionality of Pin 9.<br>00 = TACH4 (Default)<br>01 = $\overline{\text{THERM}}$<br>11 = $\overline{\text{SMBALERT}}$<br>10 = GPIO |
| <2> | $\overline{\text{THERM}}$ Disable | R/W | $\overline{\text{THERM}}$ Disable = 1, disables $\overline{\text{THERM}}$ overtemperature output features. |
| <3> | Max/Full on $\overline{\text{THERM}}$ | R/W | Max/Full on $\overline{\text{THERM}}$ = 0; when $\overline{\text{THERM}}$ temperature limit is exceeded, fans go to full speed.<br>Max/Full on $\overline{\text{THERM}}$ = 1; when $\overline{\text{THERM}}$ temperature limit is exceeded, fans go to maximum programmed fan speed.<br>Max/Full on $\overline{\text{THERM}}$ = 1; when $\overline{\text{THERM}}$ limit is exceeded, fans go to maximum speed as defined in Register 0x38, Register 0x39, Register 0x3A. |
| <4> | RES<br>$\overline{\text{THERM}}$ Config | | Unused on ADT7473. On the ADT7473−1:<br>0 = Remote Channel 2 (Default)<br>1 = Remote Channel 1 and Remote Channel 2 |
| <5> | BpAtt$V_{CCP}$ | R/W | Bypass $V_{CCP}$ attenuator. When set, the measurement scale for this channel changes from 0 V (0x00) to 2.2965 V (0xFF). |
| <6> | RES | | Reserved |
| <7> | RES | | Reserved |

1. This register becomes read-only when the Configuration Register 1 Lock bit is set to 1. Any further attempts to write to this register have no effect.

**Table 95. REGISTER 0X7E − MANUFACTURER'S TEST REGISTER 1 (POWER-ON DEFAULT = 0X00)**

| Bit No. | Mnemonic | R/W | Description |
|---|---|---|---|
| <7:0> | Reserved | Read-only | Manufacturer's test register. These bits are reserved for manufacturer's test purposes and should not be written to under normal operation. |

**Table 96. REGISTER 0X7F − MANUFACTURER'S TEST REGISTER 2 (POWER-ON DEFAULT = 0X00)**

| Bit No. | Mnemonic | R/W | Description |
|---|---|---|---|
| <7:0> | Reserved | Read-only | Manufacturer's test register. These bits are reserved for manufacturer's test purposes and should not be written to under normal operation. |

**Table 97. REGISTER 0X80 − MANUFACTURER'S TEST REGISTER 3 (POWER-ON DEFAULT = 0X10)**

| Bit No. | Mnemonic | R/W | Description |
|---|---|---|---|
| <7:0> | Reserved | Read-only | Manufacturer's test register. These bits are reserved for manufacturer's test purposes and should not be written to under normal operation. |

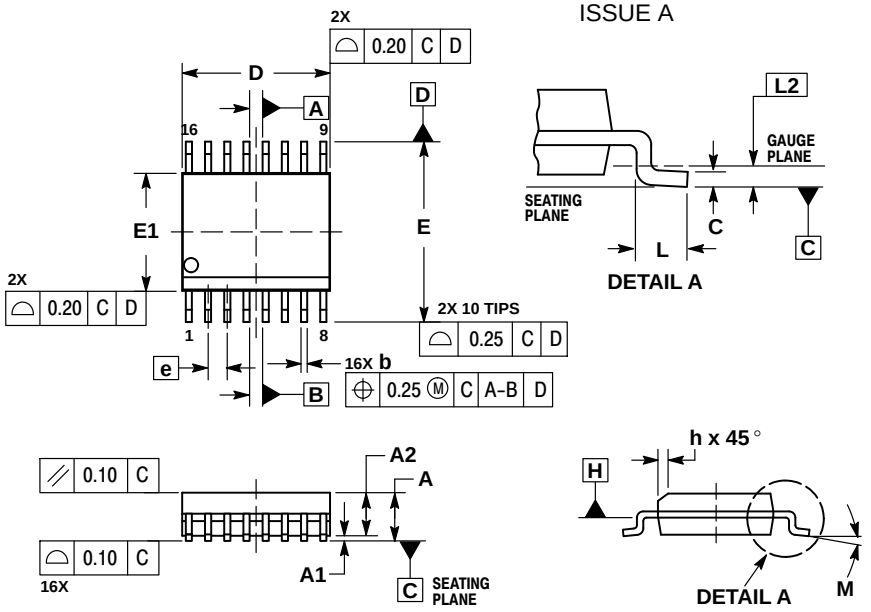
**Table 98. ORDERING INFORMATION**

| Device Order Number* | Package Type | Package Option | Shipping† |
|---|---|---|---|
| ADT7473ARQZ−REEL | 16-lead QSOP | RQ−16 | 2,500 Tape & Reel |
| ADT7473ARQZ−1RL | 16-lead QSOP | RQ−16 | 2,500 Tape & Reel |

†For information on tape and reel specifications, including part orientation and tape sizes, please refer to our Tape and Reel Packaging Specifications Brochure, BRD8011/D.
*These are Pb-Free packages.

## PACKAGE DIMENSIONS

**QSOP16**
CASE 492
ISSUE A



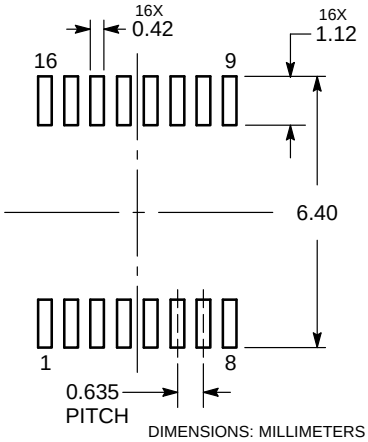




NOTES:
1. DIMENSIONING AND TOLERANCING PER ASME Y14.5M, 1994.
2. CONTROLLING DIMENSION: MILLIMETERS.
3. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.
4. DIMENSION D DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS, OR GATE BURRS. MOLD FLASH, PROTRUSIONS, OR GATE BURRS SHALL NOT EXCEED 0.005 PER SIDE. DIMENSION E1 DOES NOT INCLUDE INTERLEAD FLASH OR PROTRUSION. INTERLEAD FLASH OR PROTRUSION SHALL NOT EXCEED 0.005 PER SIDE. D AND E1 ARE DETERMINED AT DATUM H.
5. DATUMS A AND B ARE DETERMINED AT DATUM H.

| | INCHES | | MILLIMETERS | |
|---|---|---|---|---|
| DIM | MIN | MAX | MIN | MAX |
| A | 0.053 | 0.069 | 1.35 | 1.75 |
| A1 | 0.004 | 0.010 | 0.10 | 0.25 |
| A2 | 0.049 | ---- | 1.24 | ---- |
| b | 0.008 | 0.012 | 0.20 | 0.30 |
| c | 0.007 | 0.010 | 0.19 | 0.25 |
| D | 0.193 BSC | | 4.89 BSC | |
| E | 0.237 BSC | | 6.00 BSC | |
| E1 | 0.154 BSC | | 3.90 BSC | |
| e | 0.025 BSC | | 0.635 BSC | |
| h | 0.009 | 0.020 | 0.22 | 0.50 |
| L | 0.016 | 0.050 | 0.40 | 1.27 |
| L2 | 0.010 BSC | | 0.25 BSC | |
| M | 0 ° | 8 ° | 0 ° | 8 ° |

## SOLDERING FOOTPRINT




DIMENSIONS: MILLIMETERS

*For additional information on our Pb-Free strategy and soldering details, please download the ON Semiconductor Soldering and Mounting Techniques Reference Manual, SOLDERRM/D.

Pentium is a registered trademark of Intel Corporation.



### PUBLICATION ORDERING INFORMATION

**LITERATURE FULFILLMENT**:
Literature Distribution Center for ON Semiconductor
19521 E. 32nd Pkwy, Aurora, Colorado 80011 USA
**Phone**: 303−675−2175 or 800−344−3860 Toll Free USA/Canada
**Fax**: 303−675−2176 or 800−344−3867 Toll Free USA/Canada
**Email**: orderlit@onsemi.com

**N. American Technical Support**: 800−282−9855 Toll Free USA/Canada
**Europe, Middle East and Africa Technical Support:**
Phone: 421 33 790 2910
**Japan Customer Focus Center**
Phone: 81−3−5817−1050

**ON Semiconductor Website**: **www.onsemi.com**

**Order Literature**: http://www.onsemi.com/orderlit

For additional information, please contact your local Sales Representative