	−	
	±	
	±	
		°
	−	°

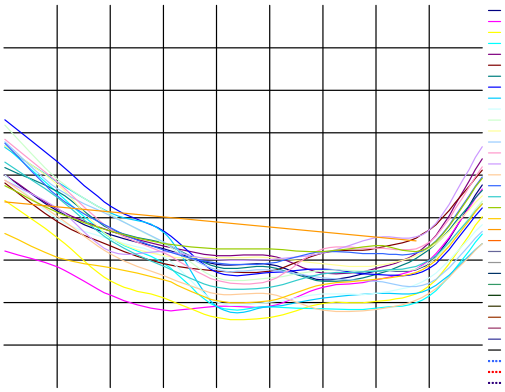
	∓	–	–		
	∓ ∓	– –	– –	∘ ≥ ≥ ∘ ∘ ≥ ≥ ∘ –	

		–	– –	–	
		– –	– –	–	
		–		–	

		×	–	–	
		–	–		

		–	±	–	μ
		–	±	–	μ
		–		–	

			±



The ADT7490 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 1), and an input line for the serial clock (Pin 2). All control and programming functions for the ADT7490 are performed over the serial bus. In addition, Pin 14 can be reconfigured as an $\overline{\text{SMBALERT}}$ output to signal out-of-limit conditions.

The ADT7490 is pin and register map compatible with the ADT7476A. The new or additional features are detailed in the following sections.

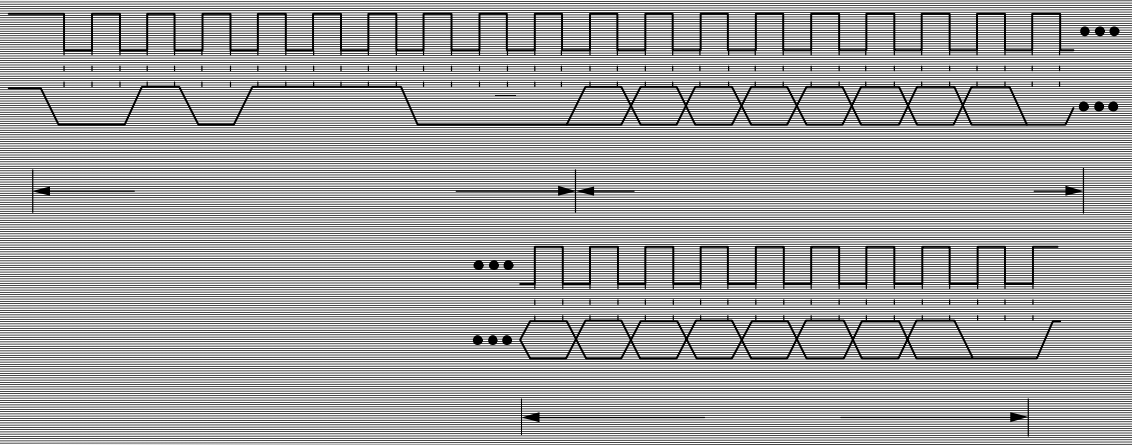
CPU thermal information is provided through the PECI input. The ADT7490 has PECI master capabilities and can read the CPU thermal information through the PECI

the I_{MON} value and the measured V_{CCP} value on Pin 23, the CPU power consumption can be calculated. The I_{MON} information can be considered as an early indication of an increase in CPU temperature.

At startup, the ADT7490 turns the fans on to 100% PWM. This allows the most robust operation at turn-on.

Control of the ADT7490 is carried out using the serial system management bus (SMBus). The ADT7490 is connected to this bus as a slave device, under the control of a master controller. The ADT7490 has a 7-bit serial bus address. When the device is powered up with Pin 13 (PWM3/$\overline{\text{ADDREN}}$) high, the ADT7490 has a default SMBus address of 0101110 or 0x2E. The read/write bit must be added to obtain the 8-bit address. If more than one ADT7490 is to be used in a system, each ADT7490 is placed in address select mode by strapping Pin 13 low on powerup. The logic state of Pin 14 then determines the device's SMBus address. The logic of these pins is sampled on powerup.

The device address is monitored from powerup but not latched until 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 $\overline{\text{ADDREN}}$/$\overline{\text{ADDR SELECT}}$ pins. Any attempted changes in the address have no effect after this.

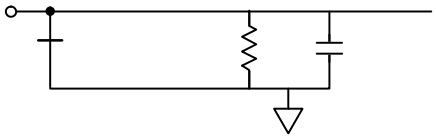


For the ADT7490, the send byte protocol is used to write

output of ON Semiconductor’s VR11.1 controllers. I_{MON} is a voltage representation of the CPU current.

All analog inputs are multiplexed into the on-chip, successive approximation, analog-to-digital converter. This ADC has a resolution of 10 bits. The basic input range is 0 V to 2.25 V, but the inputs have built-in attenuators to allow measurement of 2.5 V, 3.3 V, 5.0 V, 12 V, and the processor core voltage V_{CCP} without any external components. To allow the tolerance of these supply voltages, the ADC produces an output of 3/4 full scale (768 decimal or 0x300 hexadecimal) for the nominal input voltage, and therefore, has adequate headroom to cope with overvoltages.

The internal structure for the analog inputs is shown in Figure 24 . The input circuit consists of an input protection diode, an attenuator, plus a capacitor to form a first-order low-pass filter that gives input immunity to high frequency noise.



A number of other functions are available on the ADT7490 to offer the system designer increased flexibility. The functions described in the following sections are

The ADT7490 has four temperature measurement channels: one local, two remote thermal diodes, and a PECI. The local and thermal diode readings are analog temperature measurements, whereas PECI is a digital temperature reading.

The PECI interface is a dedicated thermal interface. The CPU temperature measurement is carried out internally in the CPU. This information is digitized and transferred to the ADT7490 via the PECI interface. The ADT7490 is a PECI host device and therefore, polls the CPU for thermal information.

The PECI measurement differs from traditional thermal diode temperature measurements in that the measurement is a relative value instead of an absolute value. The PECI reading is a negative value that indicates how close the CPU temperature is

Each PECI channel also has an associated status bit to indicate if the PECI high or low limits have been exceeded. An alert is generated on the $\overline{\text{SMBALERT}}$ pin when these status bits are asserted.

The REPLACE mode is configured by setting Bit 4 of Register 0x36. In this mode, the data in the existing Remote 1 registers are replaced by PECI0 data. This is a legacy mode that allows the thermal data from CPU1 to be stored in the same registers as in the ADT7476A. This reduces the software changes in systems transitioning from CPUs with thermal diodes to CPUs with a PECI interface. However, note that even though the associated registers are swapped, the correct data format (PECI vs. absolute temperature, see Table 6) must be written to and interpreted from these registers.

In Table 16, registers listed under the Remote 1 Default column are in absolute temperature format by default and are in PECI format in REPLACE mode. Registers listed under the PECI0 Default column are in PECI format by default and in absolute temperature format in REPLACE mode.

○	
○	
○	
○	
○	
○	

switched between I and $N1 \times I$, giving ΔV_{BE1}, and then between I and $N2 \times I$, giving ΔV_{BE2}. The temperature can then be calculated using the two ΔV_{BE} measurements. This method can also cancel the effect of any series resistance on

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.

$$\Delta = - / \times (+)$$

To factor this in, the user can write the ΔT value to the offset register. The ADT7490 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 ADT7490, I_{HIGH}, is 192 μA and the low level current, I_{LOW}, is 12 μA. If the ADT7490 current levels do not match the current levels specified by the CPU manufacturer, it may 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 ADT7490, the best accuracy is obtained by choosing devices according to the following criteria:

- Base-emitter voltage greater than 0.25 V at 12 μA at the highest operating temperature.
- Base-emitter voltage less than 0.95 V at 192 μA at the lowest operating temperature.
- Base resistance less than 100 Ω.
- Small variation in h_{FE}

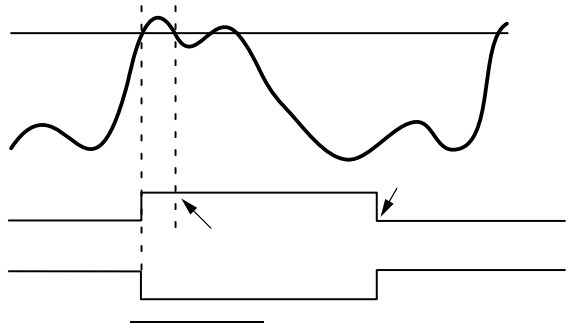
When Bit 7 of Configuration Register 6 (0x10) is set, the

interrupt has cleared. Interrupt status register bits are sticky. Whenever an interrupt 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 interrupt status bit is to read the interrupt status register after the event has gone away. Interrupt status mask registers allow individual interrupt sources to be masked from causing an $\overline{\text{SMBALERT}}$ on the dedicated alert pin. However, if one of these masked interrupt sources goes out of limit, its associated interrupt status bit is set in the interrupt status registers.

Full details of the Interrupt Status and Interrupt Mask registers associated with each measurement channels are detailed in the Table 27 and in the full register map in the Register Tables section.

The ADT7490 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.



The $\overline{\text{SMBALERT}}$ interrupt function is disabled by default. Pin 10 or Pin 14 can be reconfigured as an $\overline{\text{SMBALERT}}$ output to signal out-of-limit conditions.

Pin 14 on the ADT7490 has three possible functions: $\overline{\text{SMBALERT}}$, $\overline{\text{THERM}}$, and TACH4. The user chooses the required functionality by setting Bit 0 and Bit 1 of Configuration Register 4 at Address 0x7D.

If $\overline{\text{THERM}}$ is enabled (Bit 1, Configuration Register 3 at Address 0x78),

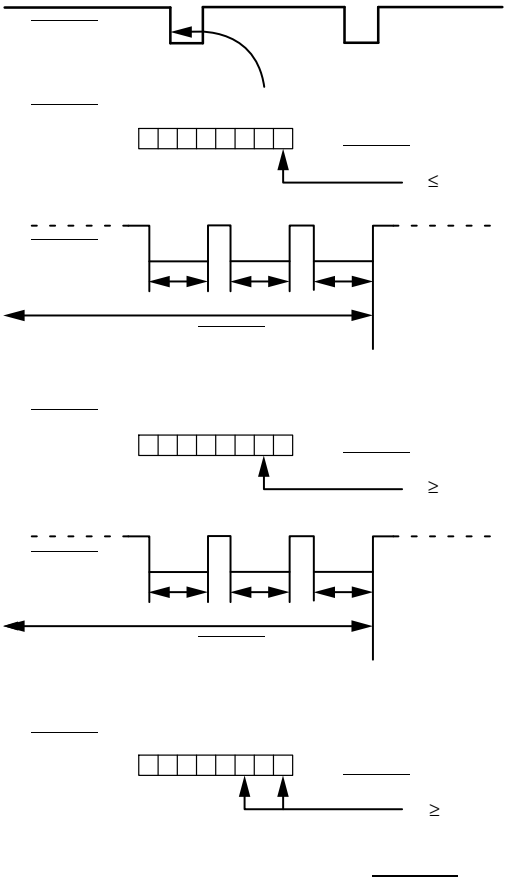
- Pin 22 becomes $\overline{\text{THERM}}$.
- If Pin 14 is configured as $\overline{\text{THERM}}$ (Bit 0 and Bit 1 of Configuration Register 4 at Address 0x7D), $\overline{\text{THERM}}$ is enabled on this pin.

If $\overline{\text{THERM}}$ is not enabled,

- Pin 22 becomes a 2.5 V_{IN} measurement input.
- If Pin 14 is configured as $\overline{\text{THERM}}$, $\overline{\text{THERM}}$ is disabled on this pin.

When $\overline{\text{THERM}}$ is configured as an input, the user 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.

The user can also set up the ADT7490 so that the fans run at 100% when the $\overline{\text{THERM}}$



When using the THERM timer, be aware of the following:

After a THERM timer read (Register 0x79):

- The contents of the timer are cleared on read.
- Bit 5 of Interrupt Status 2 register (0x42) needs to be cleared (assuming that the THERM timer limit has been exceeded).

If the THERM timer is read during a THERM assertion, the following happens:

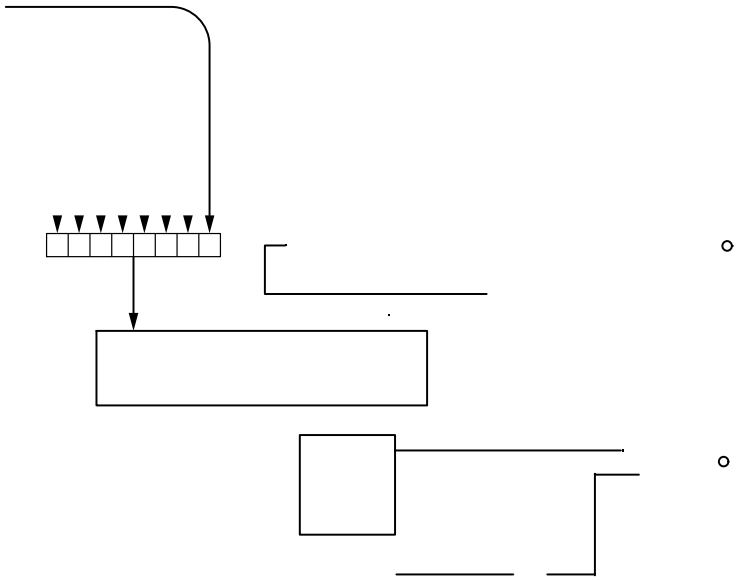
- The contents of the timer are cleared.
- Bit 0 of the THERM timer is set to 1, because a THERM assertion is occurring.
- The THERM timer increments from zero.
- If the THERM timer limit (Register 0x7A) = 0x00, the F4P bit is set.

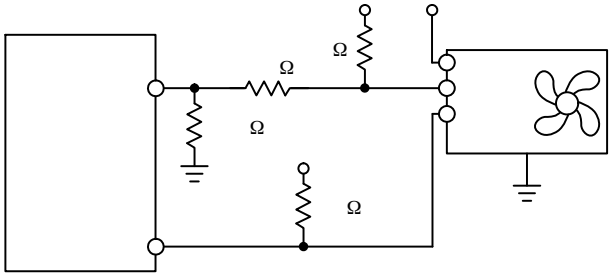
The ADT7490 can generate SMBALERTs when a programmable THERM timer limit has been exceeded. This allows the system designer to ignore brief, infrequent THERM assertions while capturing longer THERM timer events. Register 0x7A is the THERM timer limit register. This 8-bit register allows a limit from 0 sec (first THERM assertion) to 5.825 sec to be set before an SMBALERT is generated. The THERM timer value is compared with the contents of the THERM timer limit register. If the THERM timer value exceeds the THERM timer limit value, the FAN4 bit (Bit 5) of Status Register 2 is set and an SMBALERT is generated.

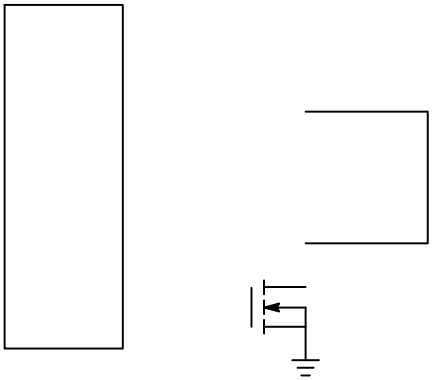
Note that depending on which pins are configured as a THERM timer, setting the FAN4/THERM bit (Bit 5) of the Interrupt Mask Register 2 (0x75), or bit 0 of the Interrupt Mask Register 1 (0x74), masks out SMBALERT; although the FAN4 bit of Interrupt Status Register 2 is still set if the THERM timer limit is exceeded.

Figure 34 is a functional block diagram of the THERM timer, THERM limit, and its associated circuitry. Writing a value of 0x00 to the THERM Timer Limit register (0x7A) causes an SMBALERT to be generated on the first THERM assertion. A THERM18 NET366.917 582.962 TD.3928 Tw[(F)74.3(AN4)

BIOS can 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 >2.914 sec in Hour 3, this indicates that system performance is degrading significantly because $\overline{\text{THERM}}$ is asserting more frequently on an hourly basis.







The ADT7490 has four TACH inputs available for fan speed measurement, but only three PWM drive outputs. If a fourth fan is being used in the system, it should be driven from the PWM3 output in parallel with the third fan.

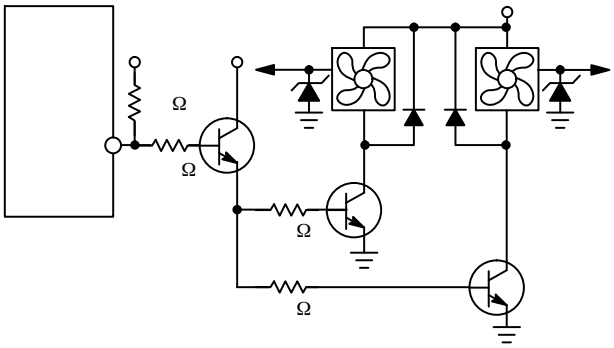
Figure 39 shows how to drive two fans in parallel using low cost NPN transistors. Figure 40 shows the equivalent circuit using a MOSFET.

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 outputs are not required to source current, and that they sink less than the 5 mA maximum current specified in the data sheet.

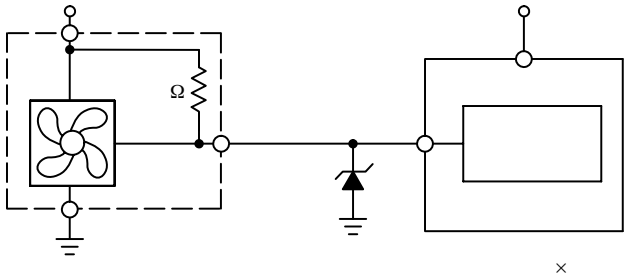
TACH measurements for fans are synchronized to particular PWM channels; for example, TACH1 is synchronized to PWM1. TACH3 and TACH4 are both synchronized to PWM3, so PWM3 can drive two fans. Alternatively, PWM3 can be programmed to synchronize TACH2, TACH3, and TACH4 to the PWM3 output. This allows PWM3 to drive two or three fans. In this case, the drive circuitry looks the same, as shown in Figure 39 and Figure 40. The SYNC bit in Register 0x62 enables this function.

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

Bit 4 (SYNC) = 1, synchronizes TACH2, TACH3, and TACH4 to PWM3.

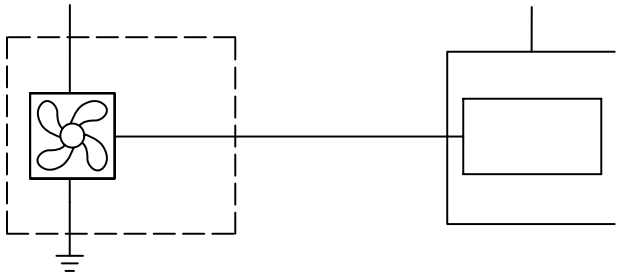


If the fan output has a resistive pullup to 12 V, or other voltage greater than 3.6 V, the fan output can be clamped with a Zener diode, as shown in Figure 43. The Zener diode voltage should be chosen so that it is greater than V_{IH} of the TACH input but less than 3.6 V, allowing for the voltage tolerance of the Zener. A value of between 3.0 V and 3.6 V is suitable.



×

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 44.



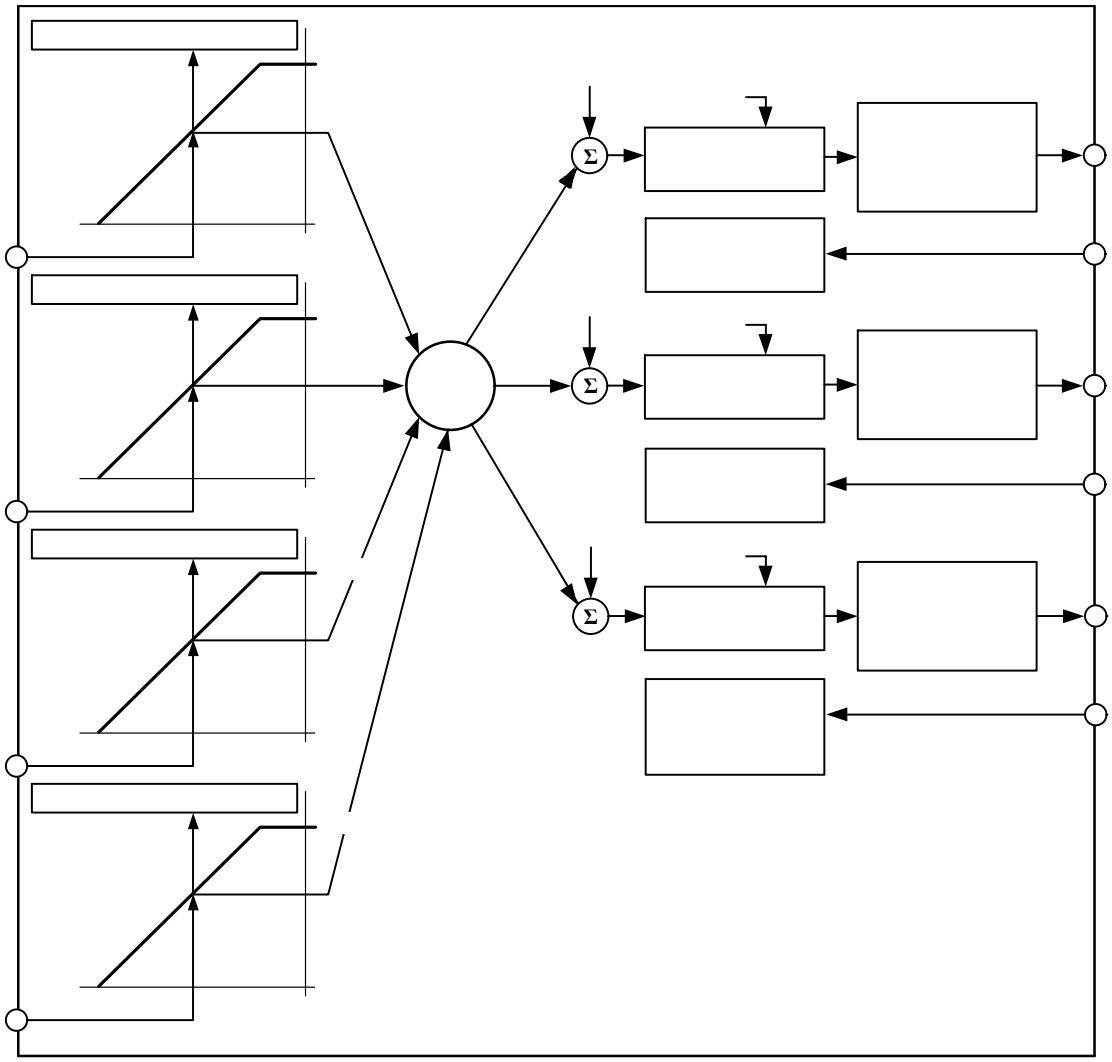
-

0xFFFF indicates that either the fan has stalled or is running very slowly (<100 RPM).

High Limit > Comparison Performed

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

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



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. Ask the following questions:

- What ADT7490 functionality is used?
- PWM2 or $\overline{\text{SMBALERT}}$?
- TACH4 fan speed measurement or overtemperature $\overline{\text{THERM}}$ function?
- 2.5 V_{IN} voltage monitoring or overtemperature $\overline{\text{THERM}}$ function?

The ADT7490 offers multifunctional pins that can be reconfigured to suit different system requirements and physical layouts. These multifunction pins are software programmable.

- How many fans are 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.
- Is the CPU fan to be controlled using the ADT7490, or will the CPU fan run at full speed 100% of the time?

If run at 100%, it frees up a PWM output, but the system is louder.

- Where is the ADT7490 going to be physically located in the system?

This influences the assignment of the temperature measurement channels to particular system thermal zones. For example, locating the ADT7490 close to the VRM controller circuitry allows the VRM temperature to be monitored using the local temperature channel.

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, can be run manually (under software control), or can be run 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, respectively. The values selected for these bits determine how the multiplexer connects a temperature measurement channel to a PWM output.

Bits [7:5] (BHVR), Register 0x5C, Register 0x5D, and Register 0x5E, with the ALT bit (Bit 3) cleared to 0.

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 channels controls PWMx

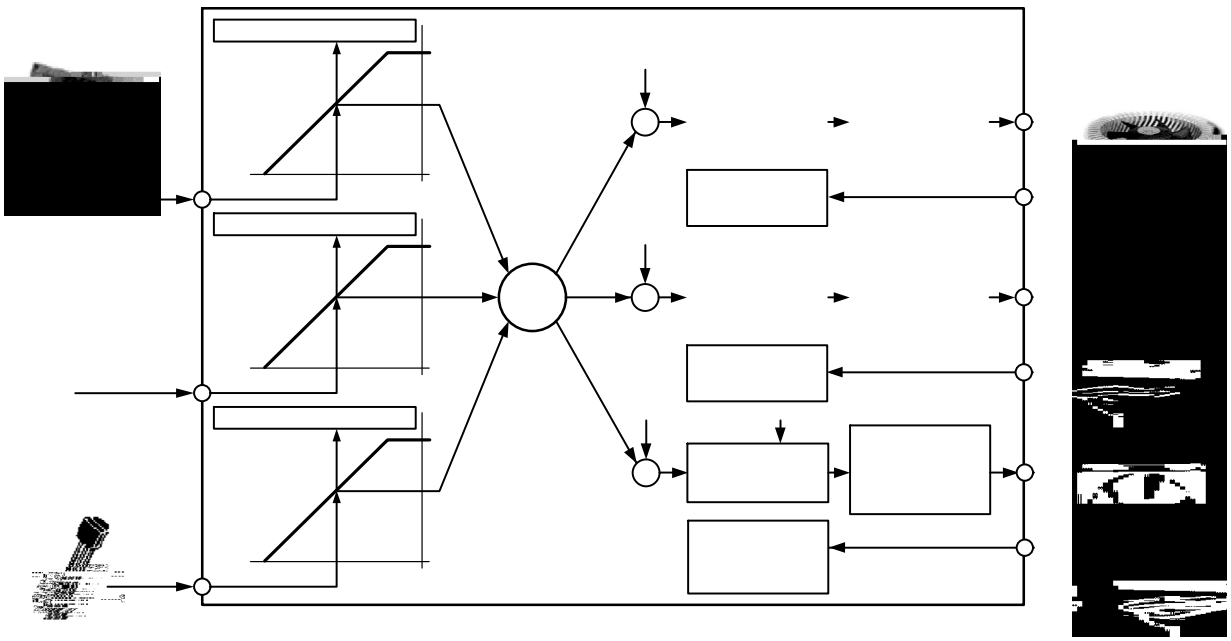
The fastest speed calculated options pertain to controlling one PWM output based on multiple temperature channels. The thermal characteristics of the three temperature zones can be set to drive a single fan. An example is the fan turning on when the Remote 1 temperature exceeds 60°C or if the local temperature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

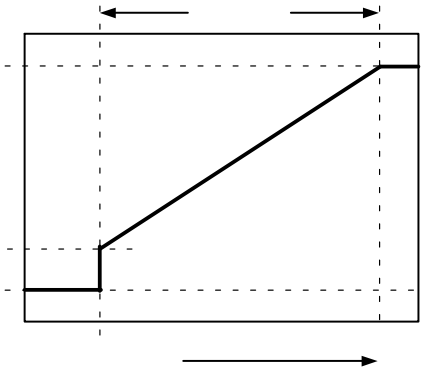
Setting the ALT bit in Register 0x5C, Register 0x5D, and Register 0x5E gives alternative behavior settings for Bits [7:5] of the PWM configuration registers.

Bits [7:5] (BHVR), Register 0x5C, Register 0x5D, and Register 0x5E, with the ALT bit (Bit 3) set to 1.

000 = PECI0 reading controls PWMx
001 = PECI1 reading controls PWMx
010 = PECI2 reading controls PWMx
011 = PECI3 reading controls PWMx
101 = Fastest speed calculated by all four PECI readings controls PWMx
111 = Fastest speed calculated by all thermal zones (Local, Rem1, Rem2 and PECI) controls PWMx

Bits [7:5] (BHVR),



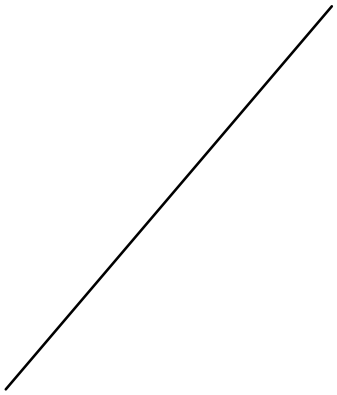


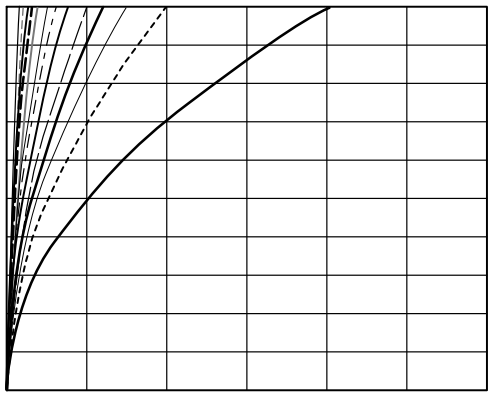
The T

While the automatic fan control algorithm describes the general response of the PWM output, it is also necessary to note that the enhanced acoustics registers (0x62, 0x63, and 0x3C) can be used to set/clamp the maximum rate of change of PWM output for a given temperature zone. This means that if T_{RANGE} is programmed with an AFC slope that is quite steep, a relatively small change in temperature could cause a large change in PWM output and possibly an audible change in fan speed, which can be noticeable/annoying to end users.

Decreasing the speed of the PWM output changes by programming the smoothing on the appropriate temperature channels (Register 0x62 and Register 0x63) changes how fast the fan speed increases/decreases in the event of a temperature spike. Slowly the PWM duty cycle increases until the PWM duty cycle reaches the appropriate duty cycle as defined by the AFC curve.

Figure 57 shows PWM duty cycle vs. temperature for each T_{RANGE} setting. Figure 57B shows how each T_{RANGE} setting affects fan speed vs. temperature. As can be seen from the graph, the effect on fan speed is nonlinear.





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 one should ensure that it is not exceeded under normal system operating conditions.

Note that $T_{\overline{THERM}}$ limits are non-maskable and affect the fan speed no matter how 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).

		°
		°
		°
		°

$\overline{THERM}$ hysteresis on a particular channel is configured via the hysteresis settings in the following section (0x6D and 0x6E). For example, setting hysteresis on the Remote 1 channel also sets the hysteresis on Remote 1 $\overline{THERM}$.

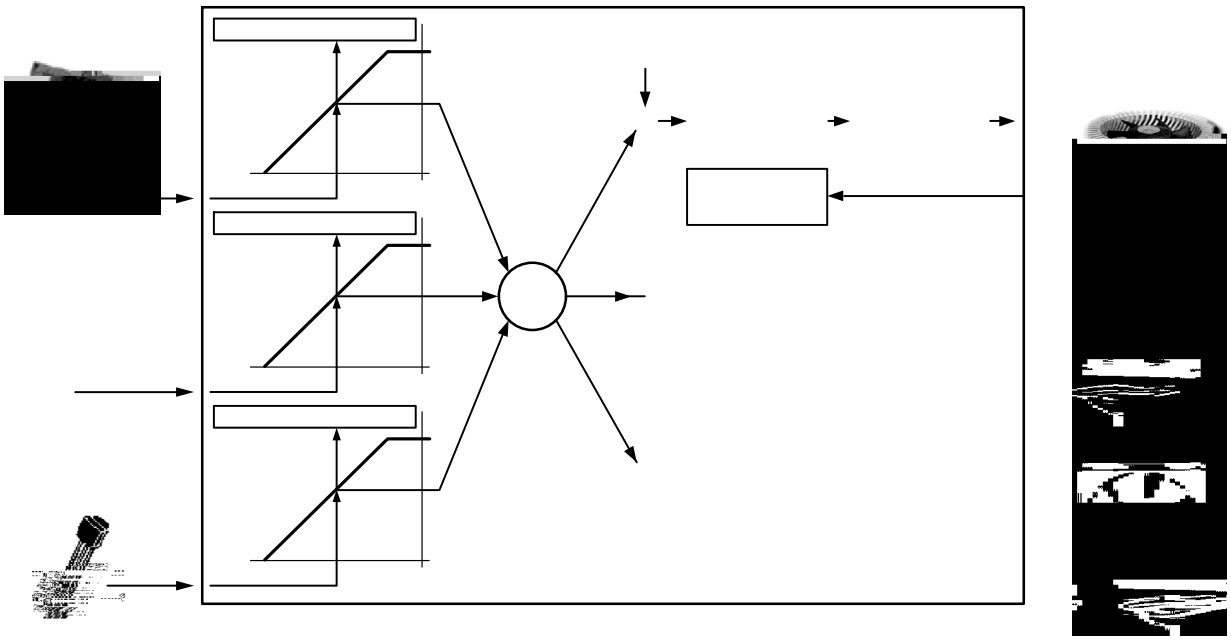
Register 0x6D, Remote 1, Local Hysteresis Register

Bits [7:4], Remote 1 Temperature Hysteresis (4°C default)
Bits [3:0], Local Temperature Hysteresis (4°C default)

Register 0x6E, Remote 2, PECI Temperature Hysteresis Register

Bits [7:4], Remote 2 Temperature Hysteresis (4°C default)
Bits [3:0], PECI 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 ever be programmed to 0°C, because this disables hysteresis. In effect, this causes the fans to cycle (during a $\overline{THERM}$ event) between normal speed and 100% speed, or, while operating close to T_{MIN}, between normal speed and off, creating unsettling acoustic noise.



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

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

Bit 0 (SLOW) = 1, slows the ramp rate for PWM changes associated with the Remote 1 temperature channel by 4.

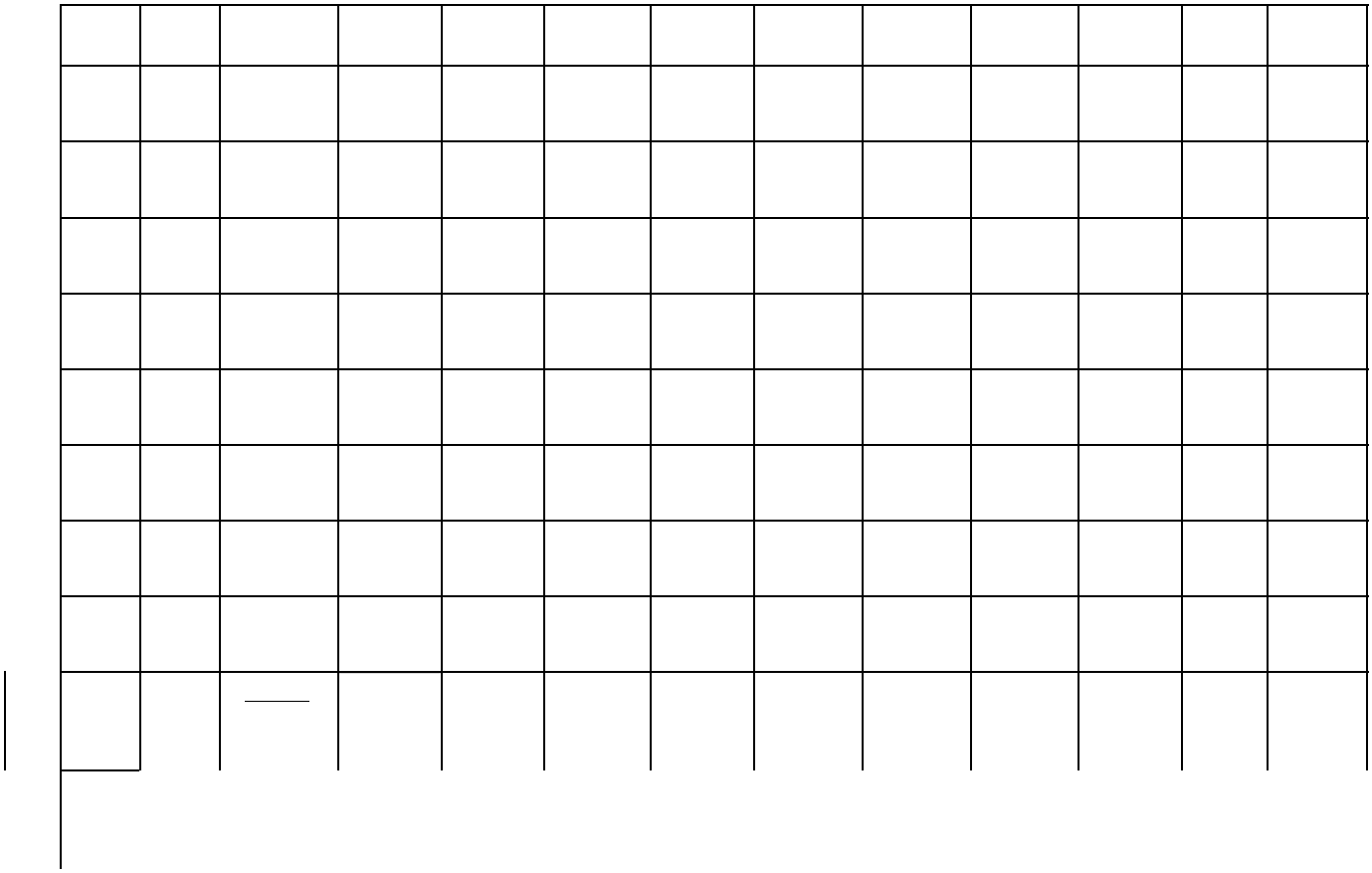
Bit 1 (SLOW) = 1, slows the ramp rate for PWM changes associated with the local temperature channel by 4.

Bit 2 (SLOW) = 1, slows the ramp rate for PWM changes associated with the Remote 2 temperature channel by 4.

Bit 7 (ExtraSlow) = 1, slows the ramp rate for all fans by a factor of 39.2%.

The following sections list the ramp-up times when the SLOW bit is set for each temperature monitoring channel.

Bits [2:0] ACOU, selec



–		–	–									

–

– °

– °

µ

–

			× × ×
			μ

–

–

–

–

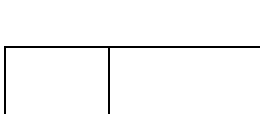
	———		

-

–

-



			○
			○
			○

			○
			○
			○

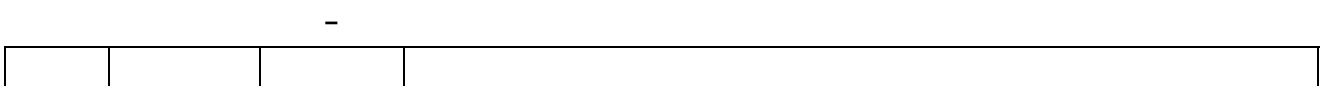
–

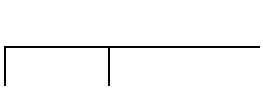
–

–

–

–





–

–

–

		_ ◦ ◦ ◦

–

		_ ◦ ◦ ◦

–

		_ ◦ ◦ ◦

–

		_ ◦ ◦ ◦

–		–	

QSOP24 NB
CASE 492B−01
ISSUE A

DATE 06 MAY 2008

SCALE 2:1