Fan-Speed Control Techniques in PCs

by David Hanrahan

Analog Devices offers a comprehensive set of hardware-monitoring products for use in desktop and notebook PCs and servers. Intelligent systems-monitoring devices make possible sophisticated fan speed control techniques to provide adequate cooling and maintain optimal thermal performance in the system. During the past year a family of products has been developed, including the ADM1029 Dual PWM Fan Controller and Temperature Monitor, the ADM1026, and ADM1030/ADM1031 Complete, ACPI-Compliant, Dual-Channel ±1°C Remote Thermal Monitor with integrated fan controller for one or two independent fans. They build on the core technology used in the ADM102x PC System Monitor product portfolio (see also *Analog Dialogue* 33-1 and 33-4). Providing fan speed control based on the temperatures measured within the system, these new products offer more complete thermal-management solutions. We discuss here the need for this level of sophisticated control and the issues inherent in providing it.

BACKGROUND

As the new millennium dawns, processors are achieving speeds of 1 GHz and more. Their impressive improvements in speed and system performance are accompanied by the generation of increasing amounts of heat within the machines that use them. The need to safely dissipate this heat, along with moves in the computing industry to develop "Green PCs" and user-friendly machines (as Internet appliances become mainstream), has driven the need for, and development of, more sophisticated cooling and thermal management techniques.

PCs have also begun to become smaller and less conventional in size and shape—as can be seen in any of the latest concept PCs or slim-line notebooks on the market. Rigid power dissipation specifications such as "Mobile Power Guidelines '99" (Ref. 1) stipulate how much heat may be safely dissipated through a notebook's keyboard without causing user discomfort. Any excess heat must be channeled out from the system by other means, such as convection along heat pipes and a heat-spreader plate, or the use of a fan to move air through the system. Clearly, what is needed is an intelligent, effective approach to thermal management that can be universally adopted. Various industry groups have assembled to address these and other issues, and have developed standards such as ACPI (advanced configuration and power interface) for *notebook PCs* and IPMI (intelligent platform management interface) for *server* management.

INDUSTRY STANDARDS

The development of the new thermal management/speed control products was motivated by the *ACPI* and *IPMI* standards.

The *Advanced Configuration and Power Interface*—ACPI (Ref 2) was defined by Intel, Microsoft, and Toshiba primarily to define and implement power management within notebook PCs.

Power management (Ref. 2) is defined as "Mechanisms in hardware and software to minimize system power consumption, manage system thermal limits, and maximize system battery life. Power management involves trade-offs among system speed, noise, battery life, processing speed, and ac power consumption."

Consider first a notebook-PC user who types trip reports while flying across oceans or continents. Which characteristic is more important, maximum CPU performance or increased battery life? In such a simple word-processor application, where the time between a user's keystrokes is almost an eternity in CPU clock cycles, maximum CPU performance is nowhere near as critical as continuous availability of power. So CPU performance can be traded off against increased battery life. On the other hand, consider the user who wants to watch the latest James Bond movie in full-motion, full-screen, mind-numbing sound and brightness, on digital versatile disk (DVD). It is critical that the system operates at a level of performance to decode the software fast enough, without dropping picture or audio frames. In this situation CPU performance cannot be compromised. Therefore, heat generation will be at top levels, and attention to thermal management will be of paramount importance to obtain top performance without impairing reliability. Enter ACPI.

What then is ACPI? ACPI is a specification that describes the interface between components and how they behave. It is not a purely software or hardware specification since it describes how the BIOS software, OS software, and system hardware should interact.

The ACPI specification outlines two distinct methods of system cooling: *passive cooling* and *active cooling*. Passive cooling relies on the operating-system (OS) and/or basic input/output-system (BIOS) software to reduce CPU power consumption in order to reduce the heat dissipation of the machine. How can this be achieved? By making intelligent decisions such as entering *suspend* mode if no keystroke or other user interaction has been detected after a specified time. Or if the system is performing some intensive calculations, such as 3D processing, and is becoming dangerously hot, the BIOS could decide to throttle (slow down) the CPU clock. This would reduce the thermal output from the machine, but at the cost of overall system performance. What is the benefit of this *passive*-type cooling? Its distinct advantage is that the system power requirement is lowered silently (fan operation is not required) in order to decrease the system temperature, but this does limit performance.

So, what about active cooling? In an actively cooled system, the OS or BIOS software takes a direct action, such as turning on a CPU mounted fan, to cool down the processor. It has the advantage that the increased airflow over the CPU's metal slug or heat-sink allows the heat to be drawn out of the CPU relatively quickly. In a passively cooled system, CPU throttling alone will prevent further heating of the CPU, but the thermal resistance of the heat sink to "still air" can be quite large, meaning that the heat sink would dissipate the heat to the air quite slowly, delaying a return to full-speed processing. Thus, a system employing active cooling can combine maximum CPU performance and faster heat dissipation. However, operation of the fan introduces acoustic noise into the system's environment and draws more power. Which cooling technique is better? In reality, it depends on the application; a

versatile machine will use both techniques to handle differing circumstances. ACPI outlines the cooling techniques in terms of two different modes: *performance mode* and *silent mode*. The two modes are compared in Figures 1 and 2.
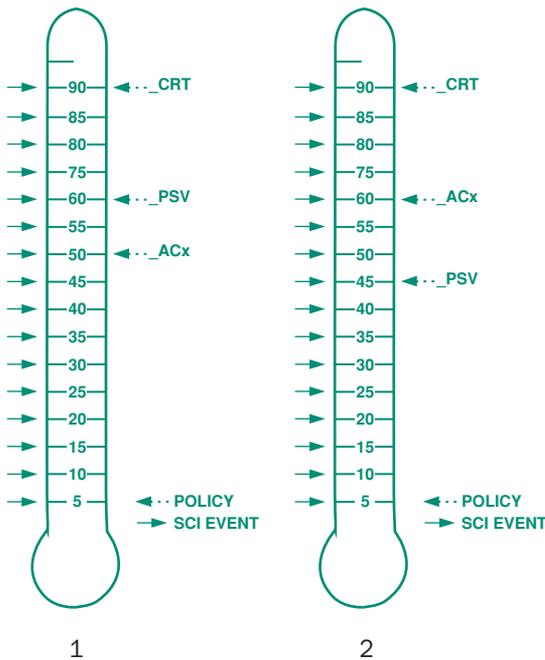

Figure 1. Performance preferred. Active mode (_ACx, fan on) is entered at 50°, passive mode (_PSV, throttle back) is entered at 60°. Shutdown occurs at the critical temperature (_CRT) 90°. Fan speed may increase at levels above ACx.
Figure 2. Silence and battery economy preferred. Passive mode is first entered at 45°, and fan is not turned on until 60°.

Figures 1 and 2 are examples of temperature scales that illustrate the respective trade-offs between performance, fan acoustic noise, and power consumption/dissipation. In order for a system-management device to be ACPI-compliant, it should be capable of signaling limit crossings at, say, 5°C intervals, or *SCI* (system-control interrupt) *events*, that a new *out-of-limit* temperature increment has occurred. These events provide a mechanism by which the OS can track the system temperature and make informed decisions as to whether to throttle the CPU clock, increase/decrease the speed of the cooling fan, or take more drastic action. Once the temperature exceeds the _CRT (critical temperature) *policy setting*, the system will be shut down as a fail-safe to protect the CPU. The other two policy settings shown in Figures 1 and 2 are _PSV (passive cooling, or CPU clock throttling) and _ACx (active cooling, when the fan switches on).

In Figure 1 (performance mode), the cooling fan is switched on at 50°C. Should the temperature continue to rise beyond 60°C, clock throttling is initiated. This behavior will maximize system performance, since the system is only being slowed down at a higher temperature. In Figure 2 (silent mode), the CPU clock is first throttled at 45°C. If the temperature continues to rise, a cooling fan may be switched on at 60°C. This reduced-performance mode will also tend to increase battery life, since throttling back the clock reduces power consumption.

Figure 3 shows how the limits of the temperature measurement bands track the temperature measurement. Each limit crossing produces an interrupt.
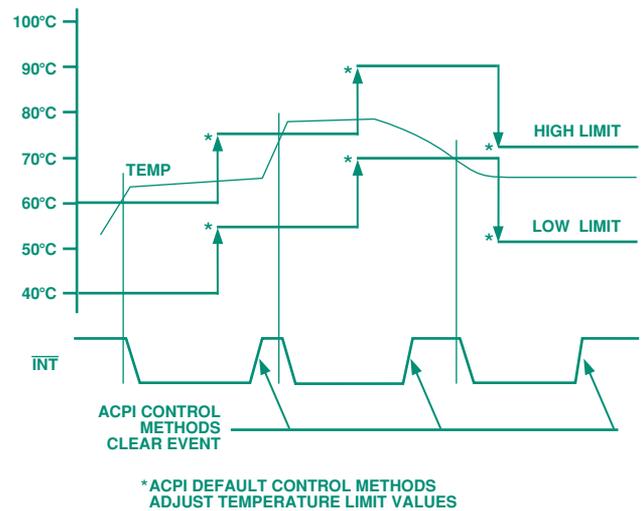

Figure 3. Tracking temperature changes by moving limits and generating interrupts.

The *intelligent platform management interface* (IPMI) specification (Ref. 3) brings similar thermal management features to *servers*. IPMI is aimed at reducing the *total cost of ownership* (TCO) of a server by monitoring the critical "heartbeat" parameters of the system: temperature, voltages, fan speeds, and PSUs (power-supply units). Another motivation for IPMI is the need for interoperability between servers, to facilitate communication between baseboards and chassis. IPMI is based on the use of a 5-V I^{2}C bus, with messages sent in packet form. Further information on IPMI is available from the Intel website at <http://developer.intel.com/design/servers/ipmi/>

All members of the Analog Devices Temperature and Systems-Monitoring (TSM) family are ACPI- and IPMI-compliant.

TEMPERATURE MONITORING
The prerequisite for intelligent fan-speed control within PCs is the ability to measure both system and processor temperature accurately. The temperature monitoring technique used has been the subject of many articles (for example, see *Analog Dialogue* 33-4) and will only be briefly visited here. All Analog Devices system-monitoring devices use a temperature monitoring technique known as *thermal diode monitoring* (TDM). The technique makes use of the fact that the forward voltage of a diode-connected transistor, operated at a constant current, exhibits a negative temperature coefficient, about –2 mV/°C. Since the absolute value of V_{BE} varies from device to device, this feature by itself is unsuitable for use in mass-produced devices because each one would require individual calibration. In the TDM technique, two different currents are successively passed through the transistor, and the voltage change is measured. The temperature is related to the difference in V_{BE} by:

$$\Delta V_{BE} = kT/q \times \ln(N)$$

where: k = Boltzmann's constant

q = electron charge magnitude

T = absolute temperature in kelvins
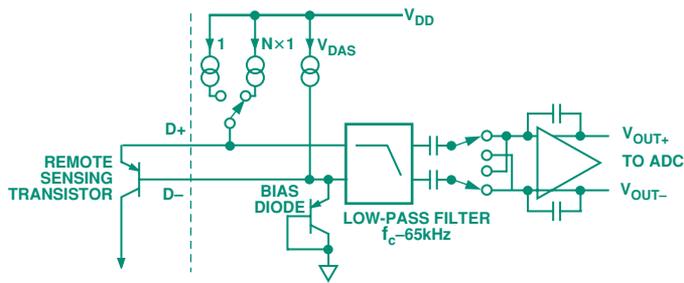
N = ratio of the two currents

Figure 4. Basic TDM signal-conditioning circuit.

In any CPU, the most relevant temperature is that of the "hot spot" on the die. All other temperatures in the system (including the heat-sink temperature) will lag the rise in this temperature. For this reason, practically every CPU manufactured since the early Intel Pentium® II processors contains a strategically located transistor on its die for thermal monitoring. It gives a true, essentially instantaneous, profile of die temperature. Figure 5 shows temperature profiles in a system repeatedly entering and waking up from *suspend* mode. It compares the temperatures measured by a thermistor attached to the CPU's heat sink and by the substrate thermal diode. In the short interval for the actual die temperature to change back and forth by about 13°, the heat sink thermistor cannot sense any change.
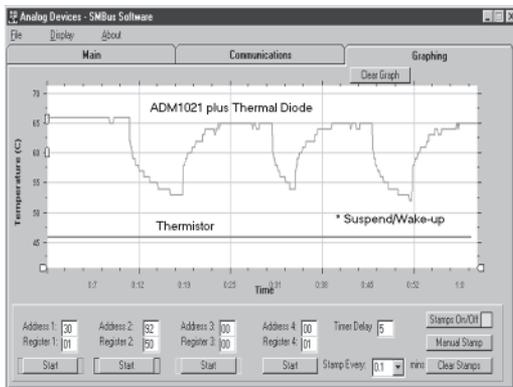

Figure 5. Comparison of temperatures measured by a heat sink thermistor and by TDM during a series of entrances to and exits from *suspend* mode.

TEMPERATURE TO FAN CONTROL

With an accurate temperature monitoring method established, effective fan control can be implemented! The technique, in general, is to use TDM to measure temperature, with the sensing transistor either integrated on-chip or externally placed as near as possible to a hot-spot, and setting the fan speed at a level that will ensure sufficient heat transport at that temperature. Various operating parameters of the control loop will be programmable, such as minimum speed, fan start-up temperature, speed versus temperature slope, and turn on/off hysteresis. The speed control approaches described will include on-off, continuous ("linear"), and pulsewidth modulation (PWM).

Fan-control methods: Historically, the range of approaches to fan-speed control in PCs is from simple on-off control to closed-loop temperature-to-fan speed control.

Two-step control: This was the earliest form of fan-speed control adopted in PCs. The BIOS would measure the system temperature (originally using a thermistor in close proximity to the CPU) and decide whether to switch a cooling fan fully on or off. Later, PCs used more accurate TDM-based temperature monitors to implement the same two-step fan control.

Three-step control: The BIOS or Operating System again measures the temperature using a thermistor or thermal diode and, based on software settings, decides whether to turn the fan fully on, fully off, or set it to run at half-speed.

Linear fan-speed control: This more recent method of fan-speed control is also known as *voltage control*. The BIOS or OS reads the temperature from the TDM measurement circuit and writes back a byte to an on-chip DAC, to set the output voltage in order to control the speed of the fan. An example of an IC fan controller of this type is the ADM1022, which has an 8-bit DAC on-chip with an output voltage range of 0 V to 2.5 V. It works with an external buffer amplifier having appropriate design ratings for the chosen fan. The ADM1022 also contains default automatic hardware trip points that cause the fan to be driven at full-speed in the event that its TDM circuit detects an over-temperature condition. The debut of these types of devices signified the emergence of *automatic fan-speed control*, where some of the decision-making is moved from OS software to system-monitoring hardware.

Pulsewidth-modulation fan-speed control: In ADI's systems-monitoring product line, these PWM types are the most recent fan control products. The BIOS or OS can read the temperature from the TDM device and control the speed of the cooling fan by adjusting the PWM duty cycle applied to it.

It is worth noting that all of the above methods of fan-speed control rely on CPU or host intervention to read the temperature from the TDM device over the 2-wire System Management Bus. The thermal management software executed by the CPU must then decide what the fan speed should be and write back a value to a register on the system's monitor IC to set the appropriate fan speed.

An obvious next step in the evolution of fan-speed control is to implement an *automatic fan-speed control loop*, which could behave independently of software and run the fan at its optimum speed for a given chip temperature. There are many benefits to such closed-loop speed control.

Once the system's monitoring device has been initialized (by loading limit registers with required parameters), the control loop is then completely independent of software, and the IC can react to temperature changes without host intervention. This feature is especially desirable when a catastrophic system failure occurs, from which the system is unable to recover. If the PC crashes, the power management software in the OS is no longer executing, which results in loss of thermal management! If the PC cannot read the temperature being measured (since the PC has crashed), it cannot be expected to set the correct fan speed to provide the required level of cooling.

The other tangible benefit of a closed-loop implementation is that it will operate the fan at the optimum speed for any given temperature. This means that both acoustic noise and power consumption are reduced. Running a fan at full speed maximizes both power consumption and acoustic noise. If the fan speed can

be managed effectively through loop optimization, running only as fast as needed for a given temperature, power drain and audible fan noise are both reduced. This is an absolutely critical requirement in battery-powered notebook PC applications where every milliampere of current, or *milliamp-second of charge*, is a precious commodity.

AUTOMATIC FAN-SPEED CONTROL LOOP

Here's how one might implement an automatic fan-speed control loop, which will measure temperature using TDM techniques and set the fan speed appropriately as a function of temperature. Programmable parameters allow more complete control of the loop. The first register value to be programmed is T_{MIN}. This is the temperature (corresponding to ACx) at which the fan will first switch on, and where fan speed control will begin. Speed is momentarily set at maximum to get the fan going, then returned to the minimum speed setting (see Figure 6). The parameter that allows control of the slope of the temperature-to-fan speed function is the range from T_{MAX} to T_{MIN}, or T_{RANGE}. The programmed values for T_{MIN} and T_{RANGE} define the temperature at which the fan will reach maximum speed, i.e., $T_{MAX} = T_{MIN} + T_{RANGE}$. Programmed temperature range is selectable: 5°C, 10°C, 20°C, 40°C, and 80°C. In order to avoid rapid cycling on and off in the vicinity of T_{MIN}, *hysteresis* is used to establish a temperature below T_{MIN}, at which the fan is turned off. The amount of hysteresis that can be programmed into the loop is 1°C to 15°C. This fan control loop can be supervised by OS software over the SMBus and the PC can decide to override the control loop at any time.
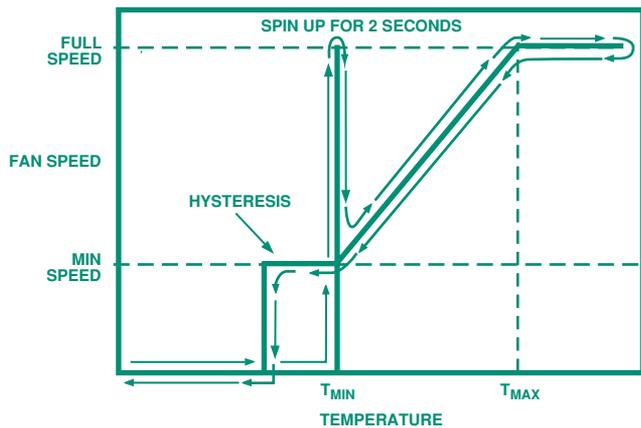

Figure 6. Fan speed programmed as an automatic function of temperature.

PWM vs. LINEAR FAN-SPEED CONTROL

One might ask why pulsewidth modulation is desirable if linear fan-speed control is already in widespread use.

Consider a 12-V fan being driven using linear fan-speed control. As the voltage applied to the fan is slowly increased from 0 V to about 8 V, the fan will start to spin. As the voltage to the fan is further increased, the fan speed will increase until it runs at maximum speed when driven with 12 V. Thus the 12-V fan has an effective operating window between 8 V and 12 V, with a range of only 4 V available for use in speed control.

The situation becomes even worse with the 5-V fan that would be used with a notebook PC. The fan will not start until the applied voltage is about 4 V. Above 4 V, the fan will tend to spin near full speed, so there is little available speed control between 4 V and 5 V. Thus, linear fan-speed control is unsuitable for controlling most types of 5 V fans.

With pulsewidth modulation, maximum voltage is applied for controlled intervals (the *duty cycle* of a square wave, typically at 30 Hz to 100 Hz). As this duty cycle, or ratio of high time to low time, is varied, the speed of the fan will change.

At these frequencies, clean tach (tachometer) pulses are received from the fan, allowing reliable fan-speed measurement. As drive frequencies go higher, there are problems with insufficient tach pulses for accurate measurement, then acoustic noise, and finally electrical spikes corrupting the tach signal. Therefore, most PWM applications use low frequency excitation to drive the fan. The external PWM drive circuitry is quite simple. It can be accomplished with a single external transistor or MOSFET to drive the fan (Figure 7). The linear fan-speed-control equivalent, driven by an analog speed voltage, requires an op amp, a pass transistor, and a pair of resistors to set the op amp gain.
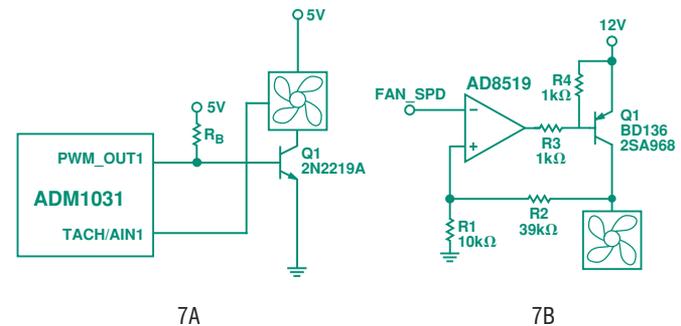
Figures 7A AND 7B. PWM drive circuit compared with a linear drive circuit.

How is the fan speed measured? A 3-wire fan has a tach output that usually outputs 1, 2, or 4 tach pulses per revolution, depending on the fan model. This digital tach signal is then directly applied to the tach input on the systems-monitoring device. The tach pulses are not counted–because a fan runs relatively slowly, and it would take an appreciable amount of time to accumulate a large number of tach pulses for a reliable fan speed measurement. Instead, the tach pulses are used to gate an on-chip oscillator running at 22.5 kHz through to a counter (see Figure 8). In effect, the tach *period* is being measured to determine fan speed. A high count in the tach value register indicates a fan running at low speed (and vice versa). A limit register is used to detect sticking or stalled fans.
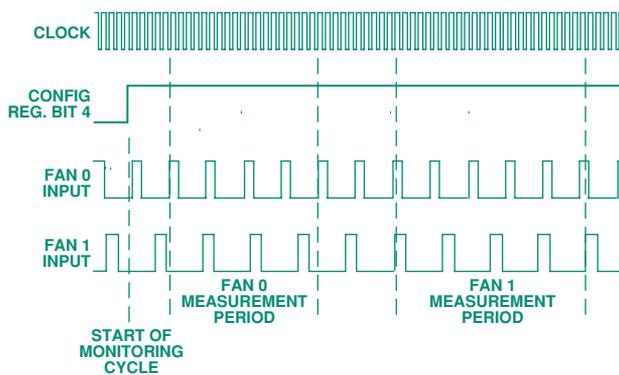
Figure 8. Fan-speed measurement.

What other issues are there with fan-speed control?

When controlling a fan using PWM, the minimum duty cycle for reliable continuous fan operation is about 33%. However, a fan will not start up at 33% duty cycle because there is not enough power available to overcome its inertia. As noted in the discussion of Figure 6, the solution to this problem is to spin the fan up for two seconds on start-up. If the fan needs to be run at its minimum speed, the PWM duty cycle may then be reduced to 33% after the fan has spun up, and it is protected from stalling by the hysteresis.

FAN STALLS AND FAN FAILURES

Nevertheless, the possibility can arise that a fan may stall at some time while used in a system. Causes may include a fan operating too slowly, or dust build-up preventing it from spinning. For this reason, the Analog Devices systems monitors have an on-chip mechanism based on the fan's tach output to detect and restart a stalled fan. If no tach pulses are being received, the value in the Tach Value register will exceed the limit in the Tach Limit Register and an error flag will be set. This will cause the controller

to attempt to restart the fan by trying to spin it up for two seconds. If the fan continues to fail, for up to five attempted restarts, a catastrophic fan failure is acknowledged to exist, and a FAN_FAULT pin will assert to warn the system that a fan has failed. In two-fan dual-controller systems, the second fan can be spun up to full speed to try to compensate for the loss in airflow due to the failure of the first fan.

SUMMARY

Superior thermal-management solutions continue to be developed and offered to the computing industry by Analog Devices. The techniques developed for the ADM1029, ADM1030/ADM1031, and ADM1026 take thermal management within PCs to a new level. These devices are packed with features such as temperature monitoring, automatic temperature control in hardware, fan-speed measurement, support for backup and redundant fans, *fan-present* and *fan-fault* detection, programmable PWM frequency, and duty cycle. As power guidelines become more stringent, and PCs run significantly hotter, more sophisticated temperature-measurement and fan-speed-control techniques are being developed to manage the systems of the future more effectively.

REFERENCES

1. *Mobile Power Guidelines '99* Revision 1.00, Intel Corporation. http://developer.intel.com/design/mobile/Intelpower/mpg99r1.pdf

2. *Advanced Configuration and Power Interface Specification*, Revision 1.0b, Intel Corporation, Microsoft Corporation, Toshiba Corp. http://www.teleport.com/~acpi/DOWNLOADS/ACPIspec10b.pdf

3. Intelligent Platform Management Interface: "What is IPMI?" Intel Corporation. http://developer.intel.com/design/servers/ipmi/ipmi.htm