Technical notes on using Analog Devices DSPs, processors and development tools
Visit our Web resources http://www.analog.com/ee-notes and http://www.analog.com/processors or e-mail processor.support@analog.com or processor.tools.support@analog.com for technical support.

Using the On-Chip Thermal Diode on Analog Devices Processors

Contributed by Mitesh Moonat

Rev 2 – August 2, 2010

Introduction

As a processor's speed increases, heat dissipation also increases. This results in an increased operating temperature. For processors that operate at higher speeds, thermal management is important to ensure reliable operation and longer life. ADSP-214xx SHARC® processors incorporate an on-chip *thermal diode* that can be used to monitor the die temperature and take appropriate actions if it goes beyond a threshold value.

The on-chip thermal diode discussed throughout this EE-Note applies to ADSP-214xx SHARC processors only (for ADSP-2147x processors, the thermal diode is only available in the 100-LQFP_EP package. See the device data sheet for more details[3]). No other Analog Devices processors from the existing portfolio today incorporate an on-chip thermal diode.

This EE-Note describes how the on-chip thermal diode can be interfaced with a *digital temperature sensor*. Example code is also provided in the associated `.ZIP` file for the thermal diode and the ADM1032 interface on the various ADSP-214xx evaluation boards.

Thermal Diode on SHARC Processors

The thermal diode on ADSP-214xx processors is a grounded collector, PNP bipolar junction transistor (BJT). The `THD_P` pin is connected to the emitter and the `THD_M` pin is connected to the base of the transistor. These pins should be connected to the `D+` and `D-` pins of the sensor being used. For details on the thermal diode, refer to the *ADSP-2146x SHARC Processor Hardware Reference*[1].

Digital Temperature Sensor

Analog temperature sensing techniques such as using thermistors, resistance temperature detectors (RTDs), and thermocouples require external signal conditioning and an analog-to digital converter (ADC). In contrast, a number of silicon-based temperature-sensing ICs output precise digital representations of the temperatures they are measuring. These are called *digital temperature sensors*. Most of them connect to a remote diode present on the chip whose temperature is to be measured, and use is based on following principle:

"Change in the V_{BE} (base-to-emitter voltage) when the diode is operated at two different currents is directly related to the absolute temperature".

The following equation shows this relationship:

$$\Delta V_{BE} = \eta_f \frac{KT}{q} \ln(N)$$

Equation 1. Change in the VBE when the diode is operated at two different currents

Where:

- K is Boltzmann's constant (1.38 × 10–23)
- q is the charge on the electron (1.6 × 10–19 Coulombs)
- T is the absolute temperature in Kelvin
- N is the ratio of the two currents
- η_f is the ideality factor of the thermal diode

Examples of these sensors include the ADM1032 device from ON Semiconductor[8] and the LM86 device from National Semiconductor[9].

In addition to measuring the remote temperature, some digital temperature sensors also have option to measure the local temperature. The sensor has a set of registers that can be accessed via a serial protocol (e.g., SMBus). These registers can be used by the host or the device measuring its own temperature for various purposes (e.g., reading local and remote temperatures, setting various high/low temperature limits, or configuring various parameters of the sensor like the conversion rate).

A digital temperature sensor also has output signals to indicate various over/under temperature conditions so that appropriate actions can be taken. The action might be to switch on a fan or lower the CPU speed, or even shut down the processor in case of a critical situation.

For example, the ADM1032 sensor has two output signals (`/THERM` and `/ALERT`). `/THERM` is a non-maskable alarm signal normally used to indicate that the temperature has exceeded a critical limit and that the system needs to be shut down. `/ALERT` is used as a warning signal which is used to take appropriate action to reduce the temperature thus preventing the system shutdown. Both `/THERM` and `/ALERT` have their own registers to set an upper temperature limit. Usually, the `/THERM` limit is higher than the `/ALERT` limit. In addition to the over-temperature alarm, the `/ALERT` signal can also be used as an indication of under-temperature or a fault condition (remote sensor open/short circuit).

On the other hand, in some systems, the `/THERM` signal can also be used to take actions other than shutting down the system (e.g., switching on/off a fan or lowering the processor clock after a particular temperature threshold is crossed). This can take advantage of the programmable hysteresis value that prevents the `/THERM` signal to de-assert as soon as the temperature goes below the upper threshold. In such systems, the `/ALERT` limit can be set higher than the `/THERM` limit to indicate that the rise in temperature cannot be controlled, even after switching on a fan or lowering the processor clock and a more drastic action is required (like shutting down the system itself).

For details about the `/THERM` and the `/ALERT` signals, refer to the ADM1032 data sheet[8]. It should also be noted that these are a few examples of how these signals can be used in various systems; system designers have the flexibility to use these signals to best their suit system requirements.

Interfacing Digital Temperature Sensors with SHARC Processors

There are two ways in which a digital temperature sensor can be interfaced with ADSP-214xx SHARC processors, depending upon whether the system has a host controller.

System with Host Controller

Some systems have a dedicated controller/host that takes care of communicating with the sensor and taking various actions, such as shutting down the system or switching on/off a fan. In this case, the sensor must connect to the `THD_P` and `THD_M` pins of the thermal diode on ADSP-214xx processor. Figure 1 shows the block diagram of such a system. The sensor has two pins (`D+` and `D-`) that connect to the remote thermal diode, two SMBus pins (`SDATA` and `SCLK`) to communicate with the host, and two outputs (`/THERM` and `/ALERT`) as temperature alarms. All the four pins (`SDATA`, `SCLK`, `/THEM`, and `/ALERT`) require external pull-up resistors (typically 10K Ohms). The `/THERM` output can be connected to a fan control circuit, and the `/ALERT` output can be connected as an interrupt to the host. `/ALERT` can be used by the host as an additional over-temperature alarm (with a different threshold than the `/THERM` signal). In addition, it can also be used to detect under-temperature, short, and open circuit conditions.

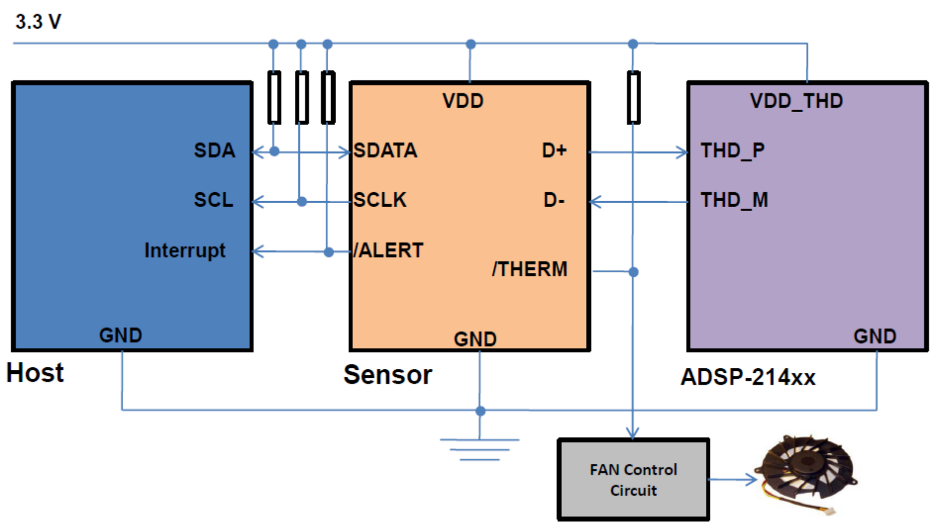

Figure 1. Digital temperature sensor interfaced with an ADSP-214xx processor and a host controller

System without a Host Controller

In systems that do not have a separate host controller, the ADSP-214xx processor communicates with the sensor itself to measure its own temperature and respond to the temperature alarms. Figure 2 shows the block diagram of such a system. The `SDATA` and `SCLK` pins from the sensor can be connected to the Two Wire Interface (TWI) of the ADSP-214xx processor (I2C® compatible). The `/ALERT` pin can go to any of the external interrupt pins such as `/IRQ0/1/2` or a DAI/DPI pin (configured to generate an interrupt). The `/THERM` signal can be connected directly to a fan control circuit. Similar to the system with a host, `/ALERT` can be used by the ADSP-214xx processor as an additional over-temperature alarm (with different limits than the `/THERM` signal). In addition, it can also be used to detect under-temperature, short, and open circuit conditions.

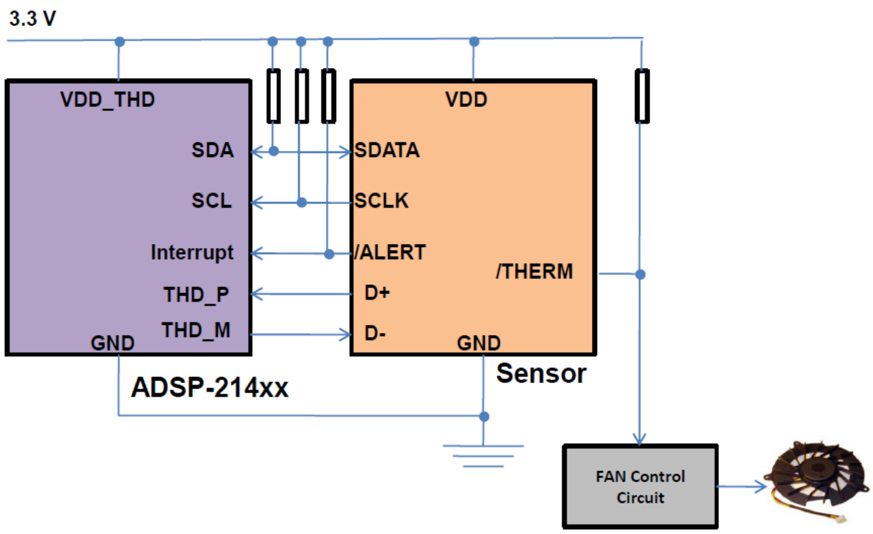

Figure 2. Digital temperature sensor interfaced with an ADSP-214xx processor without a host controller

ADM1032 on the ADSP-214xx EZ-KIT Lite® Evaluation Systems

The ADSP-21469 and ADSP-21489 EZ-KIT boards are possible configuration examples without a host controller, where the SHARC processor under measurement itself communicates with the sensor over the I2C interface. The evaluation boards illustrate all the required connections between the ADM1032 sensor and the ADSP-214xx processor. As shown in the board schematics[5][6], `VDD_THD` is connected to `VDDEXT` = 3.3V. `D+` and `D-` of the ADM1032 sensor are connected to `THD_P` and `THD_M`, while `SCLK` and `SDATA` of the ADM1032 device are connected to `DPI_PB07` and `DPI_PB08` (can be connected to `TWI_DATA` and `TWI_CLK` using switch `SW3`). Also, `/THERM` is connected to an LED to indicate an over-temperature condition.

This specific setup does not apply for the ADSP-21479 EZ-KIT board[7], since this board is populated with the 196-Ball CSP_BGA package, while the thermal diode is only available in the 100-Lead LQFP_EP package[3]. Having said this, example code for the ADSP-21479 processor is also available in the associated `.ZIP` file. This code has been tested on an internal bring-up-board (BUB) used for silicon testing and debugging.

Factors Affecting the Accuracy of Temperature Measurement

There are factors that must be considered when using a digital temperature sensor to measure the die temperature of an ADSP-214xx processor using the thermal diode. Two important factors are variation in the diode parameters and the board layout. The measurement error caused by the variation in the diode parameters can be calculated and programmed into the offset register of the digital temperature sensor. The errors introduced due to the board layout can be minimized by following the recommended board design guidelines mentioned in the corresponding sensor data sheet.

Diode Parameters

Ideality Factor

From Equation 1, we can see that the temperature measurement by the sensor is directly related to the ideality factor of the remote diode. The ideality factor may be different for different diodes and are usually specified in the processor data sheet. A digital temperature sensor is usually trimmed to a particular non-ideality factor (e.g., the ADM1032 sensor is trimmed for an ideality factor of 1.008). If the ideality factor for the diode under measurement is different from 1.008, there might be an error in the remote temperature measurement. Equation 2 may be used to calculate the error introduced at a temperature T°C.

$$\mathbf{\Delta}T = \frac{(\eta_{actual} - 1.00\mathbf{8})}{1.00\mathbf{8}(273.15\,Kelvin\ + T)}$$

Equation 2. Remote temperature measurement error

This value can be written into the offset register so that the sensor adds it to the measured temperature automatically. Note, however, that the value of the offset due to ideality factor increases with temperature.

For example, if the diode's ideality factor is 1.001 and the remote actual temperature is 80°C, the temperature error would be:

$$\mathbf{\Delta}T = \frac{(1.001 - 1.00\mathbf{8})}{1.00\mathbf{8}}(273.15\,Kelvin\ + \mathbf{80}) = -2.45^{\mathbf{0}}\ C$$

Equation 3. Remote temperature measurement error example

In order to compensate for this error, the offset register should be written with the value +2.45°C.

Series Resistance (Rs)

Series resistance is another parameter that affects the temperature measurement accuracy. This includes the series resistance of the thermal diode and that of the board trace. The diode's series resistance can be found in the processor's data sheet. However, to reduce the error due to the board trace, connect the sensor to the diode as close as possible thus keeping the trace length short.

The switched current sources cause a voltage drop across the series resistance which will be seen as an offset. Because of this, the magnitude of the temperature error will depend on both the series resistance and the values of the high and low currents being switched through the transistor. It causes the sensor to report a temperature higher than the actual temperature. Unlike the diode's non-ideality factor, the error introduced by the series resistance is constant with temperature and can be calculated by the following equation:

$$\mathbf{\Delta}T = \frac{Rs\mathbf{(}I2 - I1\mathbf{)}}{\frac{\eta K}{q}\ln(N)}$$

Equation 4. Series resistance error

For example, for ADM1032, I2 = 230 µA, and I1 = 13 µA, and assuming non-ideality factor = 1.001:

$$\mathbf{\Delta}T = \frac{Rs(230 - 13)}{\frac{1.001 * 1.38x10 - 23}{1.6x10 - 19}\ln\left(\frac{230}{13}\right)} = Rs * 0.875^{0}\ C$$

Equation 5. Series resistance error example

Assuming Rs = 5 Ohms, temperature error = 4.375°C. Since the measured temperature is more than the actual temperatures, the offset register should be written with -4.375°C.

ADSP-214xx EZ-Board Example Code

A simple example provided with this EE-Note illustrates how to communicate with the ADM1032 sensor by reading/writing from/to its registers and how to use the `/THERM` and `/ALERT` signals. The code first initializes various temperature limits and the offset register to the value of that can be defined in the include file "`Thermal_Diode_ADM1032.h`" (offset is set to zero by default and can be modified per the error calculation mentioned above). It then periodically displays the remote temperature measured and outputs an appropriate message when any of the limits (`/THERM` and `/ALERT`) are crossed. The `/THERM` output is also connected to an LED on the EZ-Board which glows until the temperature (remote/local) is above the programmed limit. Figure 3 shows the output of the example code.

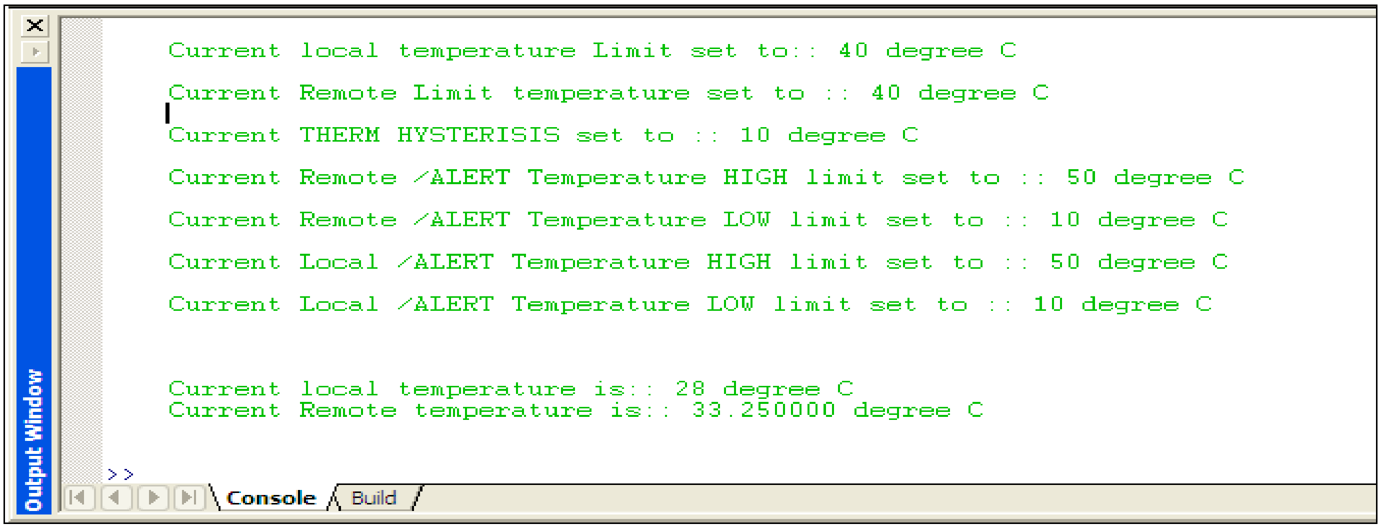

Figure 3. Output of the ADSP-21469 EZ-Board example code

References

[1] *ADSP-2146x SHARC Processor Hardware Reference*. Rev 0.2, August 2009. Analog Devices Inc.

[2] *ADSP-21469 SHARC Processor Data Sheet.* Rev 0, June 2010. Analog Devices Inc.

[3] *ADSP-21478/ADSP-21479 SHARC Processor Data Sheet.* Rev PrB, April 2010. Analog Devices Inc.

[4] *ADSP-21483/21486/21487/21488/21489 SHARC Processor Data Sheet.* Rev PrA, April 2010. Analog Devices Inc.

[5] *ADSP-21469 EZ-Board Evaluation System Manual*. Rev 1.0, April 2009. Analog Devices Inc.

[6] *ADSP-21489 EZ-Board Evaluation System Manual*. Rev 1.0, April 2010. Analog Devices Inc.

[7] *ADSP-21479 EZ-Board Evaluation System Manual*. Rev 1.0, April 2010. Analog Devices Inc.

[8] *ADM1032 data sheet.* Rev. B, January 2009, ON Semiconductor.

[9] *LM86 data sheet*. 2003, National Semiconductor.

[10] *AN1034 "Digital temperature sensor accuracy explained"*, Rev. 01, 9 August 2006, Philips.

Document History

Revision	Description
Rev 2 – August 2, 2010 *by Mitesh Moonat*	Added ADSP-2147x and ADSP-2148x processors. Tested the code examples under VisualDSP++ 5.0 Update 8.
Rev 1 – October 29, 2009 *by Mitesh Moonat*	Initial release.