

Thermal Management on PIIX4 equipped CPU boards

Francois Isabelle
Field Applications Engineering

Document revisions:
Rev A: Original release

TABLE OF CONTENT

INTRODUCTION.....	3
THERMAL MANAGEMENT	3
“I2C.....	3
SMBUS.....	3
HARDWARE IMPLEMENTATION	4
INTEL 82371AB (PIIX4) INTEGRATED SMBUS.....	4
ADM1021 SYSTEM TEMPERATURE MONITOR	6
SOFTWARE IMPLEMENTATION.....	7
BIOS IMPLEMENTATION	7
OS IMPLEMENTATION.....	7
APPLICATION IMPLEMENTATION.....	7
MORE INFORMATION	7
I2C HTTP://WWW-US2.SEMICONDUCTORS.PHILIPS.COM/I2C/	7
PIIX4 HTTP://DEVELOPER.INTEL.COM/DESIGN/CHIPSETS/DATASHTS	7

INTRODUCTION

Thermal Management

Temperature fluctuations can compromise the performance of electronic parts and systems. Failure rates can be dramatically minimized and life expectancy can increase if proper thermal management is implemented. This is why Intel implemented the System Management Bus directly in the PII X4 and this is why we encourage you to use it. This document will introduce you to thermal management capabilities.

"I2C

I2C is an acronym for Inter Integrated Circuit bus. The I2C bus was developed in the early 1980's by Philips semiconductors. Its purpose was to provide an easy way to connect a CPU to peripheral chips in a TV-set.

The BUS physically consists of 2 active wires and a ground connection. The active wires, SDA and SCL, are both bi-directional. Where SDA is the Serial Data line and SCL is the Serial Clock line.

Every component hooked up to the bus has its own unique address whether it is a CPU, LCD driver, memory, or complex function chip. Each of these chips can act as a receiver and/or transmitter depending on its functionality. Obviously a LCD driver is only a receiver, while a memory or I/O chip can both be transmitter and receiver. Furthermore there may be one or more BUS MASTER's.

The BUS MASTER is the chip issuing the commands on the BUS. In the I2C protocol specification it is stated that the IC that initiates a data transfer on the bus is considered the BUS MASTER. At that time all the others are regarded to as the BUS SLAVES. As mentioned before, the I2C bus is a Multi-MASTER BUS. This means that more than one I2C device capable of initiating data transfer can be connected to it.

SMBus

The System Management Bus is a two-wire interface through which simple power-related chips can communicate with the rest of the system. It uses I2C as its backbone. A system using SMB passes messages to and from devices instead of tripping individual control lines.

With the SMBus, a device can provide manufacturer information, tell the system what its model/part number is, save its state for a suspend event, report different types of errors, accept control parameters, and return its status.

The SMBus may share the same host device and physical bus as ACCESS bus components as long as an appropriate electrical bridge is provided between the internal SMB devices and external ACCESS bus devices."

Introduction By : nsouch@freebsd.org

Last modified: Sat Feb 20 15:48:19 CET 1999

HARDWARE IMPLEMENTATION

Intel 82371AB (PIIX4) Integrated SMBus

(Extracted from PIIX4 Documentation)

“The PIIX4 PCI function 3 contains enhanced Power Management logic with support for Device Management, Suspend and Resume states, and System Clock Control. This function also supports a System Management Bus (SMBus) Host and Slave interface. The register set associated with Power Management and SMBus controller is shown below with actual register descriptions given in section 0.”

The following table shows the SMBus related PCI configuration registers of the PIIX4. Normally, the BIOS will take care of the device configuration and the SMBBA register will be set to 0x5000. Allowing control of the SMBus master via I/O operations. For more information about PCI configuration mechanisms I recommend Mindshare's PCI System Architecture.

Table 10. PCI CONFIGURATION REGISTERS (FUNCTION 3)

Address Offset	Mnemonic	Register Name	Access
7C–7Fh	DEVRESJ	Device Resource J	R/W
80h	PMREGMISC	Miscellaneous Power Management	R/W
81–8Fh	—	Reserved	—
90–93h	SMBBA	SMBus Base Address	R/W
94–D1h	—	Reserved	—
D2h	SMBHSTCFG	SMBus Host Configuration	R/W
D3h	SMBREV	SMBus Revision ID	RO
D4h	SMBSLVC	SMBus Slave Command	R/W
D5h	SMBSHDW1	SMBus Slave Shadow Port 1	R/W
D6h	SMBSHDW2	SMBus Slave Shadow Port 2	R/W
D7–FFh	—	Reserved	—

The next tables shows the I/O register mapping of the SMBUS controller. Since I2C/SMBus is open you may find a complete specification on the internet along with a few open source implementations (Linux and FreeBSD). It is pretty straight forward and simple so this should be enough to make it run on any O/S.

Table 12. System Management Bus (SMBus) I/O Registers

Offset From Base Address	Mnemonic	Register Name	Access
00h	SMBHSTSTS	SMBus Host Status	R/W
01h	SMBSLVSTS	SMBus Slave Status	R/W
02h	SMBHSTCNT	SMBus Host Count	R/W
03h	SMBHSTCMD	SMBus Host Command	R/W
04h	SMBHSTADD	SMBus Host Address	R/W
05h	SMBHSTDAT0	SMBus Host Data 0	R/W
06h	SMBHSTDAT1	SMBus Host Data 1	R/W
07h	SMBBLKDAT	SMBus Block Data	R/W
08h	SMBSLVCNT	SMBus Slave Count	R/W
09h	SMBSHDWCMD	SMBus Shadow Command	R/W
0A–0Bh	SMBSLVEVT	SMBus Slave Event	R/W
0C–0Dh	SMBSLVDAT	SMBus Slave Data	R/W

NOTES:

1. The base address is programmable via the SMBBA Register (90–93h; function 3)

The PIIX4 Thermal Management functions provide control over the system's cpu and pci clocks. One of the clock control mode is the "thermal clock control".

"Thermal Throttle Control:

If the THRM# signal is asserted for greater than 2 seconds and the system is not in a Stop Clock, Sleep, or Deep Sleep state, PIIX4 automatically starts toggling the STPCLK# signal and ZZ signal (if [ZZ_EN] set) with a period of 244 ms and a programmable Duty Cycle.

This system toggles between full-speed operation and the Stop Grant State. The Duty Cycle can be set in 12.5% increments by programming the [THRM_DTY] bits in the Count B (CNTB) register. The functionality of thermal throttling is independent of [THRM_EN] bit, which is used to enable events for other power management functions. The [THRM_DTY] field must be programmed by the BIOS.

This emulates a reduced frequency Host clock, resulting in reduced power and thermal output. When the THRM# signal is negated, the system returns to clock control previously in use."

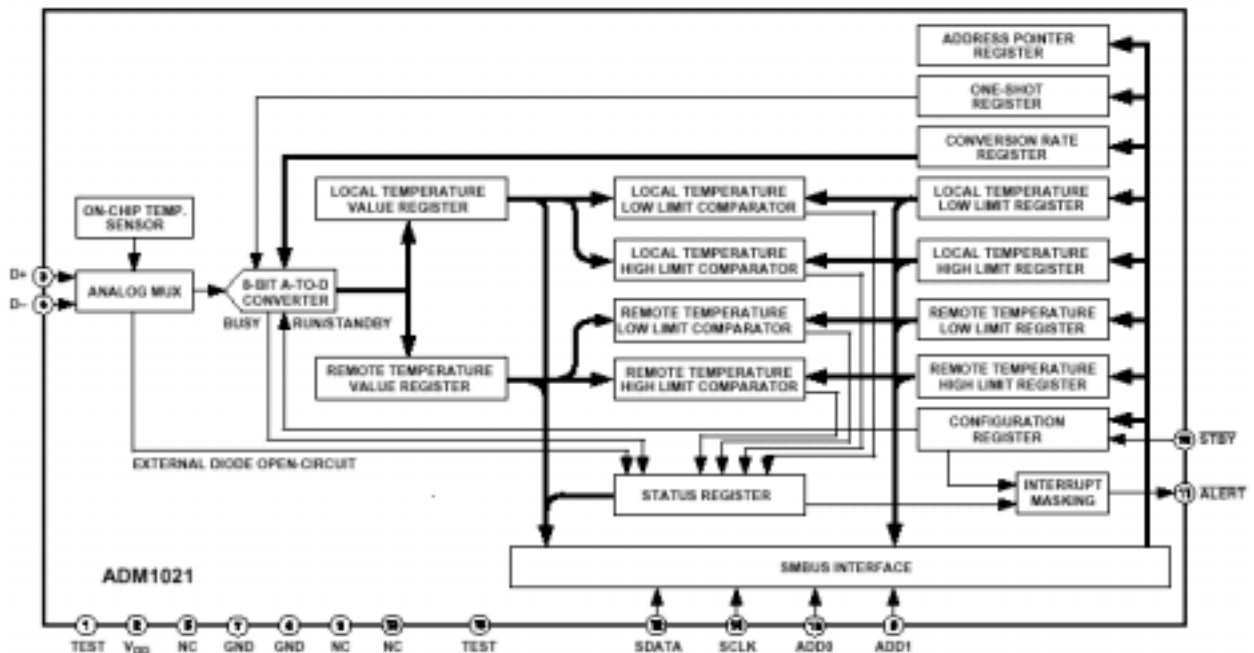
Extracted from PIIX4 specifications.

ADM1021 System Temperature Monitor

At this time we use an ADM1021 system temperature monitor or compatibles (MAX1617) on any of our PIIX4 equipped CPU board. (T941, T943 - T948). This chip will drive the PIIX4 THRM# signal if a critical thermal condition occurs. For information about the actual measurement method, you can take a look at the specifications of the Analog Devices chip:

<http://products.analog.com/products/info.asp?product=ADM1021A>.

FUNCTIONAL BLOCK DIAGRAM



Local Temperature:

Temperature taken by the on-chip temp. sensor. This is what we can call the system temperature.

Remote Temperature:

Temperature taken by the external sensor (directly on the PII-PIII cpu die). This is what we will call the cpu temperature.

Low-High Limits:

You can set the temperature range you consider safe for normal operation. If the temperature goes out of range, the device can generate an SMB alert which in turn can generate an interrupt. Then, your software can handle the situation...

SOFTWARE IMPLEMENTATION

Bios Implementation

In the Award CMOS Setup program, you can toggle the power management on and off, and you can monitor the actual readings from the sensors. Unfortunately, there is no interrupt routine wrapping this feature for MSDOS applications and other OS relying on the bios.

OS Implementation

Teknor did not develop any drivers and/or services to support thermal management. But reliable freeware implementations do exist.

Linux Hardware Monitoring is provided by LM_SENSORS, it is not especially written for PIIX4 and ADM1021 but it is highly configurable. It can autodetect the SMB devices and may run as a kernel module or directly merged within the kernel.

For more information on LM_SENSORS

<http://www.netroedge.com/~lm78/>

Choosing an OS supporting power management may be wise if you want your application to turn off the main power or change the current CPU clock on critical thermal conditions.

Application Implementation

For Windows

A popular shareware Windows application supporting a bunch of devices is available, Hardware Sensors Monitor run on Win95/Win98/WinNT/Win2000 and allows you to set a response on critical conditions event: sleep, shutdown etc... <http://www.hmonitor.com/>

For Linux

The LM_SENSORS package also includes several utilities for Linux, you can use the healtd.sh script to send an email when the system goes critical or exhibits specific behavior requiring immediate attention.. <http://www.netroedge.com/~lm78/>

MORE INFORMATION

I2C <http://www-us2.semiconductors.philips.com/i2c/>

PIIX4 <http://developer.intel.com/design/chipsets/datashts>