Analysis of Thermal Monitor features of the Intel® Pentium® M Processor

Efi Rotem, Alon Naveh, Micha Moffie and Avi Mendelson
{efraim.rotem, alon.naveh, avi.mendelson} @intel.com

Abstract

General purpose mobile processors present two new design challenges. The first is the need to achieve the best performance while fitting into limited thermal environment and the second is maximizing the battery life of the entire system. The Intel® Pentium® M processor has been developed specifically to target the mobile market and therefore, much emphasis was given to advanced mobile features which included improved power efficiency, advance ACPI interface and the Enhanced Intel SpeedStep® technolgy.

This paper focuses on both the thermal management and the battery life aspect of Intel Pentium M. The paper describes the new Enhanced Intel SpeedStep technology, evaluates its efficiency in terms of performance and energy consumption and compares thermal control schemes.

1 INTRODUCTION

Modern mobile processors are becoming thermally limited; i.e., they can run, at least some applications, in higher frequency if the amount of heat they generate would not exceed the maximum allowed thermal limitation. The thermal limitations on such systems are determined by the size of the box of the mobile system, the fan size and the noise it generates, the use of relative small heat-sinks, the heat that other platform components are generating and more. Often, Mobile platforms offer the option to trade performance in order to reduce power and energy consumption of the CPU. By doing so, the mobile platform can achieve longer battery life, lower acoustic noise and a cooler box. Thus, thermal constrains are a major design factor of mobile processors such as the Intel Pentium M processor that fit in these thin and light laptops.

Much emphasis was given, in the design of the Intel Pentium M processor, to advanced power management features which include improved power efficiency, advance ACPI interface[1] and the Enhanced Intel SpeedStep technology that targets both thermal control mechanism and energy consumption optimization. The mobile processor is expected to achieve different conflicting goals: while connecting to the AC outlet, the system is expected to achieve uncompromised performance, similar to desktop computers, while at other occasions, when it uses internal battery only, we expect the system to consume as minimal energy as possible so it can fits into the mobile thermal envelop, while achieving the maximum performance at the lowest power consumption possible.

Previous Intel® mobile microprocessors[5] used two control mechanisms. (1) A thermal monitor mechanism [6] that was based on clock throttling (referred as *TM1* in this article). It assumes that the processor can be either in a full operational mode; i.e., runs at maximum allowable frequency, or at a minimal operational mode; i.e., all clocks were stalled till the processor is cool enough. This mechanism can be activated either by operating system, via power and thermal control package such as ACPI, or by special hardware component upon identifying a hot condition. (2) A static mechanism using Voltage Scaling, which can change both the voltage as well as the frequency. While running at performance mode (either connected to AC outlet or at user preference), the CPU executed at maximum frequency and Vcc. Moving to power saving mode reduced frequency and Vcc to the minimum. This mode change was done upon request or while

the notebook user changed his usage mode.

In order to enhance the thermal control solution, The Intel Pentium M processor uses a Dynamic Voltage Scaling (DVS) based mechanism [3] [8] [9] , that can change both the voltage as well as the frequency at run time, to achieve two targets: (1) to adjust the power consumption of the system to the thermal conditions and (2) to maximize the battery life. The new mechanism that the Intel Pentium-M processor implemented, (referred to as *TM2 in this article*) can efficiently switch between different DVS voltage and frequency points to better control thermal conditions as well as achieving better energy savings.

DVS algorithms have been proposed by several papers; e.g., [9] [7] and are being used by other systems such as AMD* PowerNow!* [2] [10] and Transmeta* Longrun* technology [11] . The purpose of this paper is to describe the implementation of the Enhanced Intel SpeedStep technology in the Intel Pentium M processors, to present its measured performance and power consumption and compare different power / performance schemes.

2 ENHANCED INTEL SPEEDSTEP® TECHNOLOGY DESCRIPTION

2.1 Thermal Sensors

The Intel Pentium M processor implements 2 thermal sensors. A thermal diode connected to an external A/D on the platform, which can provide junction temperature, a digital sensor that detect the maximum allowable junction temperature, and a critical temperature indication. If the maximum allowance temperature is reached, the sensor can generate either an interrupt or a hardware based (self initiated) thermal throttle to protect the device from overheating. If the critical shutdown point is reached, the system will shutdown immediately. A block diagram is described in Figure 1.

* Other names and brands may be claimed as the property of others.

Two modes of operation are available for use on the mobile platform:

1. Software control mechanism - An external A/D continually translates the diode reading into temperature. A software control mechanism, such as ACPI [1] tracks the junction temperature and initiates linear or DVS power control based on pre-defined policy.
2. Self throttle – The digital temperature detector trips when junction temperature reaches maximum junction temperature and initiates TM1 or TM2 power control cycle. Platforms that use software control mechanisms, use self thermal monitor as a fail-safe mechanism only, since it is much faster then the software, and can guarantee fast respond time.

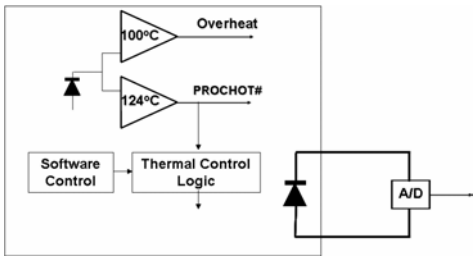

Figure 1: Thermal sensor block diagram

2.2 Enhenced Intel SpeedStep technology thermal control

The Enhanced Intel SpeedStep technology mechanism implements a Dynamic Voltage Scaling technique. An example of a DVS cycle is described in Figure 2. In this example, the CPU is operating at high Vcc and High frequency generating high power. Upon a thermal trigger which may be initiated either by the digital trip point or by a software controlled action, the CPU halts execution and locks the PLL at a new frequency. This is done within few micro seconds at the high Vcc. The frequency transition is fast enough not to interfere with the application normal flow. Once finished, the Vcc starts changing to the new value. The CPU power delivery utilized large filter capacitance and the voltage cannot be changed instantaneously. The charge and discharge rate is a function of the time

constant and, on the platform we used, was in the order of 1mV/μSec. The full power saving is achieved once the voltage transition has finished. A frequency transition up is done in reverse order; upon trigger, Vcc is ramped up first and once Vcc reaches it's final value, The CPU execution is halted for a short period of time and the frequency changed to the new higher value. An example of frequency transitions are described in Figure 2.

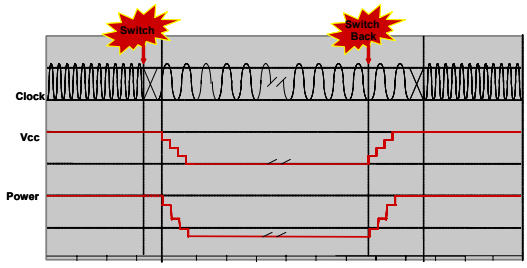

Figure 2: Dynamic Voltage Scaling cycle

The frequency transition is done fast enough to allow none interrupted application execution. As a result, DVS transitions can be utilized for energy and thermal control during the normal operation of the application flow. The target frequency and voltage are programmable by BIOS or operating system and the Intel Pentium-M processor supports multiple Vcc / Frequency points.

3 THE EXPERIMENTAL ENVIRONMENT

In order to characterize the behavior of the new Enhanced Intel SpeedStep technology, described earlier, in a thermally limited environment such as passive cooling user policy, we have built a special testing environment. The testing environment, the baseline parameters and the definition of the terminology we will use to describe the experiments are given in this section. The measured results and the characterization of the thermal control behavior, and the energy savings are described in the next section.

THERMAL CONTROL EVALUATION

The behavior of thermal management mechanism depends on the program the system is executing, on the initial ambient temperature and on the thermal control mechanism. In order to examine the system behavior, we choose to use the self trigger mechanism, since unlike software controlled mechanisms which show large variation between different runs of the same experiment, self trigger mechanisms can guarantee more predictable and more repeatable measurements. The thermal behavior was measured on a mix of traces taken form the SPEC-2K INT and SPEC-2K-Float. For the experiments described in this article, we used an Intel Pentium-M processor based mobile computer development board [12] Operating system in use was Microsoft® Windows® XP.

In order to control the thermal conditions which normally determined by the ambient temperature, cooling hardware, fan speed and other platform components, we built a special thermal control hardware which is described in Figure 3.

.

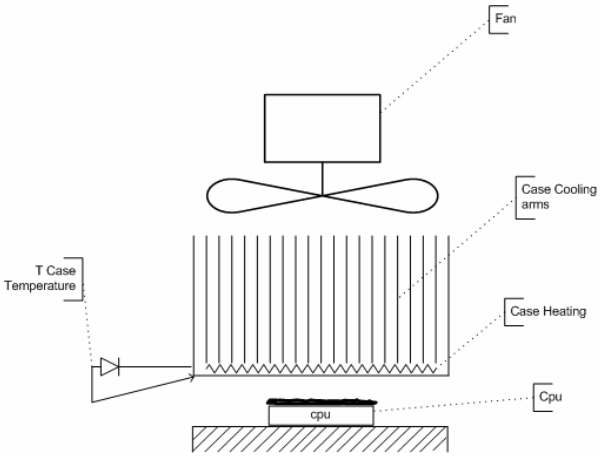

Figure 3 - Cooling/heating device

This device is intended to control the CPU case temperature. The device includes a heating plate, a cooling fan and a thermometer. The device is controlled from another computer; by pre-setting the desired target temperature of the case, the device will automatically warm/cool itself to meet the target temperature. Setting a higher case temperature is equivalent to more demanding thermal conditions and, if high enough, will trigger the thermal control mechanisms.

The Intel Pentium-M processor we used in this experiment was capable to run at frequencies between 600 to 1600MHZ in

steps of 200MHz. In addition, it was also capable to perform linear clock throttling. benchmark, where each program has an equal weight in the final graph.

4 EXPERIMENTS

4.1 Experiment 1 – Linear thermal control vs. DVS control

In the first experiment we analyzed the efficiency of the DVS (TM2) mechanism compared with clock throttling (TM1) mechanism. In this experiment we used a 1.6 GHz Intel Pentium M processor based system. The operating system thermal management features where disabled and the self trigger thermal control system were enabled to guarantee repeatable and predictable results.

We ran the SPEC-2K Integer and FP benchmarks and recorded the benchmark score at different case temperatures. We repeated the process with max frequency of 1600Mhz for TM1 and with TM2 set to the lowest frequency of 600Mhz. Figure 4 describes the self thermal control impact on benchmark scores. The same system and thermal attachment where used during the entire experiment to assure repeatable behavior.

The controlled variable in the above experiment is the case temperature. Changing the case temperature represents thermal solution capability, e.g. capability to cool different power values. Higher ambient temperature, turning fan off to save battery or more heat generated on the platform, will all reduce the cooling capability of the CPU.

Performance results, plotted on Y axis are shown as relative values. Benchmark score with no throttling (lowest T_{case}) is defined as baseline performance and considered 100% performance. Throttling causes performance degradation as shown Figure 4. Performance is plotted as percentage relative to no throttle performance. Each benchmark was run 3 times and the average score of the runs was used as the score. Figure 4 shows average of all components of SPEC 2000

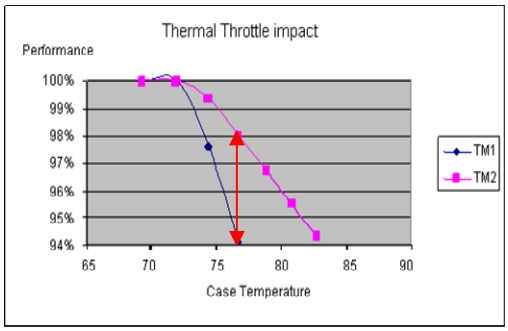

Figure 4 TM1 vs. TM2 efficiency

As case temperature rises, a more aggressive control is required, trading more performance to save the additional power. It can be noted from the results that the Performance to Power efficiency of TM2 is better then TM1 e.g. there is a need to trade less performance in order to get the same thermal performance. The slope of the charts represents the rate at which the performance needs to be reduced in order to get better thermal capability. At temperature of 77°C for example, the platform will trade 2% while using DVS and 6% performance while using linear power savings. These benefits increase as thermal control depth increases. As we go to more aggressive thermal control and willing to trade more performance, the power benefit of dynamic voltage over linear control increases. Desktop platform have sufficient cooling capabilities and therefore TM1 or TM2 are considered as fail safe mechanism. As such the performance differences are not significant. Mobile platforms, on the other hand, may trade performance to get more battery file, reduce heat to get a cooler system and reduce acoustic noise. Performance while controlling power is important and therefore DVS is used as first choice and TM1 is used only after DVS has changed to the lowest point.

4.2 Experiment 2 – DVS set-point exploration

The following experiment focused on the behavior of TM2 is different set points. Thermal control mechanism can choose different policies to compromise performance for power. We expect that for any polynomial dependency between performance and power, the optimal policy would be running the CPU at the highest frequency that still, will not exceed Tmax. In addition, the energy cost of charging and discharging big filter capacitors impact energy efficiency. In this experiment we looked at the impact of different set points on the efficiency of the control mechanism. Again, in order to guarantee predictable and repeatable results we used the hardware self trigger mechanism. We repeat the test procedure described in experiment number 1 with the same set of SPEC components. Figure 5 describes the relative performance as a function of case temperature. Overall relative performance was calculated using the same procedure as 4.1.

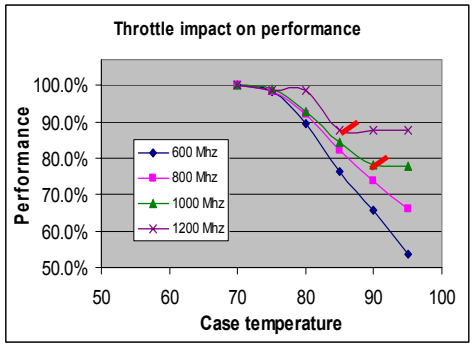

Figure 5: Control set-point impact on performance

As one could expect, out of the previous results, we can see that, as the case temperature rises, a more aggressive control is required, trading more performance to save the additional power. Please note that similar results could be observed if more power aggressive program were used on the same case temperature.

Figure 5 shows that the junction temperature of the CPU has been continuously tracked by the thermal control computer. It can be noted that 1000MHz and 1200MHz (marked with a red line) performance saturate at 85'C and 90'C case temperature respectively. At this point, the CPU runs at the target frequency but thermal control is not sufficient and junction temperature continues to increase. This temperature increase over the max Tj, was recorded by the control computer.

It can be noted that the optimal thermal control policy would bring the CPU to the highest frequency that still can cool the device. Such control algorithms have been proposed by Skadron et. al [8] and Cohen et. al [4]. The thermal equation used in Microsoft ACPI 2.0 also implements such policy.

4.3 Energy efficiency, savings and average power

Mobile computers, while operating on batteries, often trade performance for battery life. Previous generations of mobile computers offered two user selectable policies. Maximum performance, which was used while the computer is connected to the AC outlet, and maximum battery life which was used to save power while reducing frequency (performance). The Enhanced Intel SpeedStep technology offers fast and uninterrupted power state transition which enables a new policy called adaptive policy. This user selectable policy tracks the CPU workload. It increases frequency on demand, and decreases power and frequency while the CPU is at idle state for a long time.

In this experiment we measured the average power of the Intel Pentium-M processor at these 3 performance policies. We used the above mobile platform, running Microsoft® Windows® XP while having the operating system control the Enhanced Intel SpeedStep technology states. We used Mobil Mark 02 to represent mobile notebook typical workload. Power measurement was done by an external A/D card. We collected power samples at rate of 50mSec and calculated the average power over the test run.

The experiment results are described in Figure 6. The average power we measured on the Intel Pentium M processor, while running Mobile Mark 2002 was 1.1W and

benchmark score 21. Setting the notebook at maximum power saving mode reduced the power to 300mW with score of 11. This represents 66% average power saving for the cost of 48% drop in performance. The adaptive policy provides 43% power savings for only 10% performance compromise.

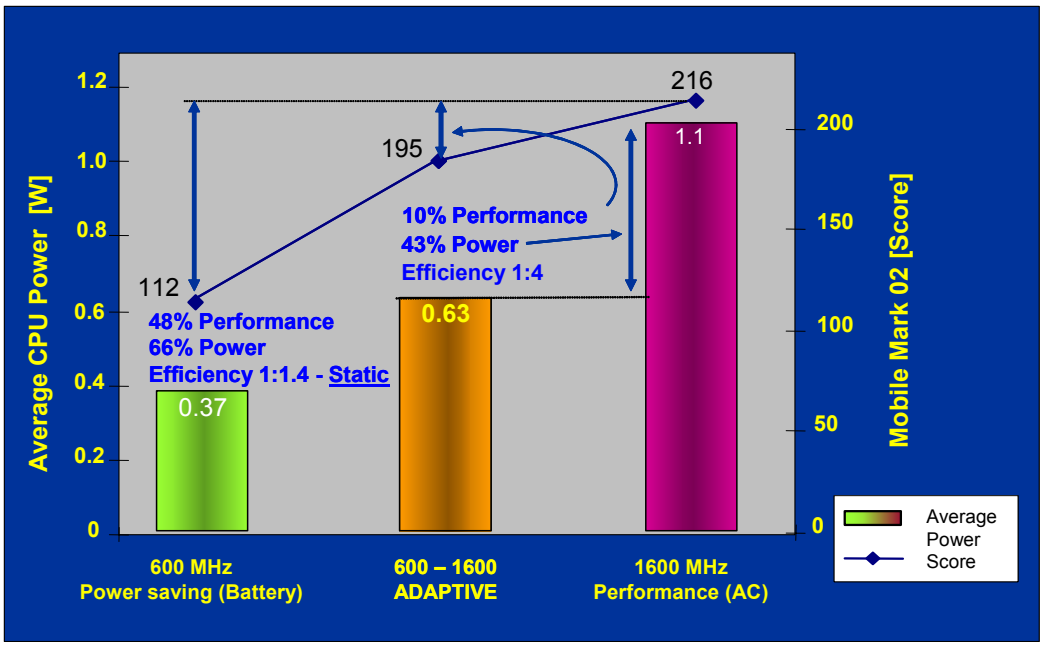

Figure 6: Average power measurements

5 SUMMARY AND CONCLUSION

The Intel Pentium M processor is the first Intel processor designed especially for the mobile market. Important vectors in this market are un-compromised performance in thermal envelope and small form factor that translate to challenging cooling requirements. The Intel Pentium-M processor introduced a new thermal monitoring mechanism, The Enhanced Intel SpeedStep Technology, based on fast respond and low overhead dynamic voltage scaling. These features enable the use of frequency and voltage scaling for both thermal control algorithms and energy savings. The evaluation of thermal control algorithms showed improved thermal capabilities with higher power / performance efficiency. We have shown that control algorithm, selecting the right control frequency can maximize processor performance, enabling cool, low noise operation mode, while keeping the device within the specified junction temperature. The enabling of the adaptive power scheme introduced a low power operation mode, saving 43% of the power with small compromise in performance.

Average power measurements showed that the best power and energy savings is achieved by lowering the CPU to it's lowest power state. Better power to performance and energy to performance scores are achieved with adaptive algorithms.

REFERENCES

[1] ACPI: Advanced Configuration & Power Interface in http://www.acpi.info

[2] AMD PowerNow Technology. - http://www.amd.com/us-en/Corporate/VirtualPressRoom/0,,51_104_857_964,00.html

[3] D. Brooks and M. Martonosi: "Dynamic thermal management for high-performance microprocessors" in Proceedings of the 7th International Conference on High Performance Computer Architecture (HPCA), January 2001

[4] Cohen A. Finkelstein L., Mendelson A., Ronen R. and Rudoy R., On Estimating Optimal Performance of CPU Dynamic Thermal Management; in IEEE computer architecture letters, Nov. 2003.

[5] Intel: Pentium®-4 Thermal management – in http://support.intel.com/support/processors/pentium4/sb/CS-007999.htm#Thermal

[6] Sassen S. : Thermal Monitor – CPU safeguard, in http://www.hardwareanalysis.com/content/article/1278.6/

[7] Semeraro, G.; Magklis, G.; Balasubramonian, R.; Albonesi, D.H.; Dwarkadas, S.; Scott, M.L in Energy-efficient processor design using multiple clock domains with dynamic voltage and frequency scaling, in High-Performance Computer Architecture, 2002. Proceedings. Eighth International Symposium on , 2-6 Feb. 2002

[8] Skadron K, Abdelzaher T, and Stan , "Control-Theoretic Techniques and Thermal RC Modeling for Accurate and Localized Dynamic Thermal Management , HPCA 2002

[9] Skadron, K.; Stan, M.R.; Wei Huang; Velusamy, S.; Sankaranarayanan, K.; Tarjan, D.; Temperature-aware computer systems: Opportunities and challenges in IEEE Micro, Volume: 23 , Issue: 6 , Nov.-Dec. 2003 Pages:52 – 61

[10] Swaminathan, V.; Schweizer, C.B.; Chakrabarty, K.; Patel, A. - A: Experiences in implementing an energy-driven task scheduler in Proceedings of RT-Linux in Real-Time and Embedded Technology and Applications Symposium, Sept. 2002 Pages:229 – 238

[11] Transmeta Longrun Technology - http://www.transmeta.com/efficeon/longrun.html

[12] Platform Design guide - http://developer.intel.com/design/mobile/desguide/252614.htm