Managing the Impact of Increasing Microprocessor Power Consumption

Stephen H. Gunther, Desktop Platforms Group, Intel Corp.
Frank Binns, Desktop Platforms Group, Intel Corp.
Douglas M. Carmean, Desktop Platforms Group, Intel Corp.
Jonathan C. Hall, Desktop Platforms Group, Intel Corp.

Index words: Pentium® 4 processor, power, power simulation, thermal management, thermal design, heat-sinks

ABSTRACT

The power dissipation of modern processors has been rapidly increasing along with increasing transistor count and clock frequencies. At the same time, there is a growing disparity between the maximum power consumption of a processor and the "typical" power consumed by that processor; i.e., power consumed while running typical applications. This trend is the result of the significant increase in transistor count required to reach the desired peak performance targets.

Designing a processor with the intent of minimizing system costs, especially those arising from high power consumption, while retaining a high level of reliability requires attention at all stages of the design. In the case of the Pentium® 4 processor, the design team focused from the beginning on reducing power consumption without negatively impacting either the performance or reliability of the processor in any significant way. Many techniques, both innovative and pre-existing, were applied across the entire processor in an effort to eliminate unnecessary power consumption. The mass adoption of these techniques resulted in a significant reduction in both maximum and typical processor power dissipation.

INTRODUCTION

The total power dissipation of recently introduced, new-generation, microprocessors has been rapidly increasing, pushing desktop system cooling technology close to its limits. The Pentium 4 processor is the first new-generation IA-32 microarchitecture processor to significantly improve upon the historical IA-32 processor power trends. The power savings achieved in the design of the Pentium 4 processor will translate into lower cost systems, higher frequency processors, and improved manufacturing yield while maintaining the high level of reliability and quality for which Intel is known.

This paper outlines the guiding principles that were set in place when the Pentium 4 processor was first defined. We first describe an engineering process and tool chain that made the low-power aspects of the design visible and supported a feedback path to the design team. We also briefly touch on the key lessons that were learned in the design, validation, and debugging of clock gating and other power-conserving elements of the Pentium 4 processor.

We then present a processor power-monitoring and control mechanism that is entirely contained on the processor die. No off-chip hardware or software interaction is required to guarantee that a pre-determined temperature threshold is not exceeded during processor operation. The architecture of this thermal monitor control logic closely maps to the existing Advanced Configuration and Power Interface (ACPI) specifications. The monitoring and control mechanism consists of three separate but related functions: a temperature detection mechanism, a power reduction mechanism, and control and visibility logic. Each of these functions, and the implementation constraints, are described in detail in this paper.

PROCESSOR POWER TRENDS

The power dissipation of modern processors is rapidly increasing as both clock frequency and the number of transistors required for a given implementation grow. Figure 1 shows the power consumption trend of processors introduced by Intel over the past 15 years. As can be seen, the general trend is for maximum processor power consumption to increase by a factor of a little more than 2X every four years.

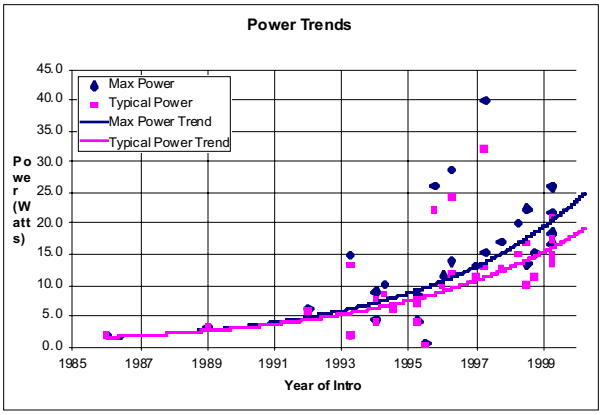

Figure 1: Trends in CPU power consumption

The second trend to note, also shown in Figure 1, is the increasing disparity between the maximum power consumption of a processor and the power consumed by that same processor while running more typical applications. For a typical Intel® processor introduced from 1996 onwards, the power consumed when running a synthetic high-power workload was 20% higher than the power consumed by the same processor while running a high-power section of a real application.

The disparity between maximum power and typical power consumed presents a particularly difficult problem to the system designer. The system must be designed to ensure that the processor does not exceed the maximum specified operating temperature, even when it is dissipating the maximum power. While designing an elaborate heat sink, or forcing more air through the system can usually meet this constraint, there is usually significant cost associated with more elaborate solutions and environmental regulations that limit system (fan) noises.

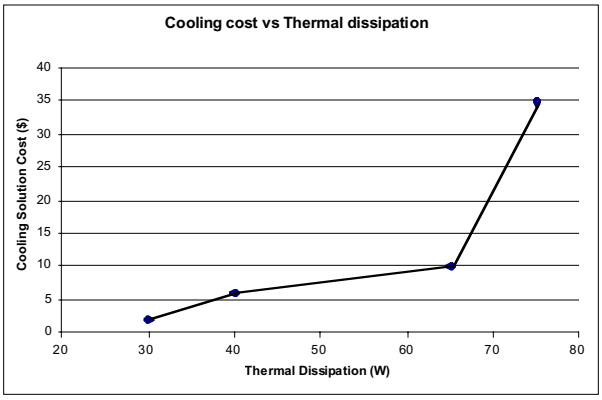

Figure 2: The cost of removing heat from a microprocessor

Figure 2 shows the relative implementation cost of various cooling solutions ranging from a simple extruded aluminum heat sink to a more elaborate heat pipe technology. It is important to observe that as power increases, there is a non-linear relationship between the cooling capabilities and the cost of the solution. This emphasizes the importance of limiting maximum power consumption to a specific envelope, one defined by the cost structure of the platform for which the processor is intended.

PROCESSOR FEATURE DEFINITION AND TRACKING

The Pentium 4 processor team started on the journey to a lower power design by defining power reduction features during the definition phase of the processor. The overall processor power reduction was to be achieved with a combination of architectural (i.e., thermal monitor) techniques and microarchitectural/circuit (i.e., clock-gating and low-power circuit) techniques. Once the power reduction features were defined, it became important to track the actual power savings of each microarchitectural feature in a manner synergistic with microprocessor development techniques.

A well-defined infrastructure was put in place to track various aspects of the Pentium 4 processor power reduction effort. The infrastructure included a series of interactive reviews, indicators, and regression tests. The first review was put in place as a checkpoint prior to the start of code development. This review examined the basic power plan and identified the specific set of power-saving features. The output of the review was a common format, easy to read summary of the power-reduction plan.

An internal design indicator, dubbed the Wattmeter, see Figure 3, was developed to track the implementation status and the power savings achieved by each feature. The Wattmeter was used to track progress toward specific power goals. The data for the Wattmeter were based on the unit-level power reduction plans, RTL coding information, and the specific circuit style utilized for each Functional Unit Block (FUB). The relative impact of each feature was combined with the expected power consumption of the FUB in question, and an architectural-level activity factor was applied to yield an estimate of the power saved by that feature in Watts. As specific features were coded, the impact of those features on power consumption was added to the Wattmeter to influence the power-savings indicator.

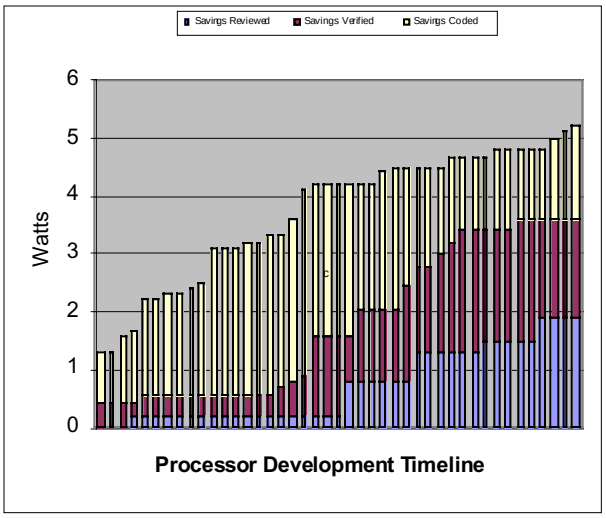

Figure 3: The Wattmeter

The Pentium 4 processor team identified more than 400 individual power-reduction opportunities that were ultimately implemented in the processor. According to projections from the various tools, some of these features resulted in a significant reduction in power, while most had a smaller impact. Power reduction features implemented in roughly 20% of the FUBs accounted for 75% of the total power savings achieved on the processor.

ARCHITECTURAL-LEVEL POWER SIMULATION

A new power estimation tool was developed to facilitate the evaluation of various power reduction features prior to the availability of a fully featured RTL model. This tool, referred to in this paper as the Architectural-Level Power Simulator (ALPS), allowed the Pentium 4 processor team to profile power consumption at any hierarchical level from an individual FUB to the full chip. The ALPS allowed power profiling of everything, from a simple microbenchmark written in assembler code, to application-level execution traces gathered on real systems.

At the most abstract level, the ALPS methodology consists of combining an energy cost associated with performing a given function with an estimate of the number of times that the specific function is executed. The energy cost is dependent on the design of the product, while the frequency of occurrence for each event is dependent on both the product design and the workload of interest. Once these two pieces of data are available, generating a power estimate is simple: multiply the energy cost for an operation (function) by the number of occurrences of that function, sum over all functions that a design performs, and then divide by the total amount of time required to execute the workload of interest.

The difficulty comes in gathering each of the required pieces of data. The benefit of being able to estimate power consumption is highest early in the design, yet detailed data on event frequency and energy cost are often not available. Therefore, it is often necessary to make significant approximations based on the data that are available.

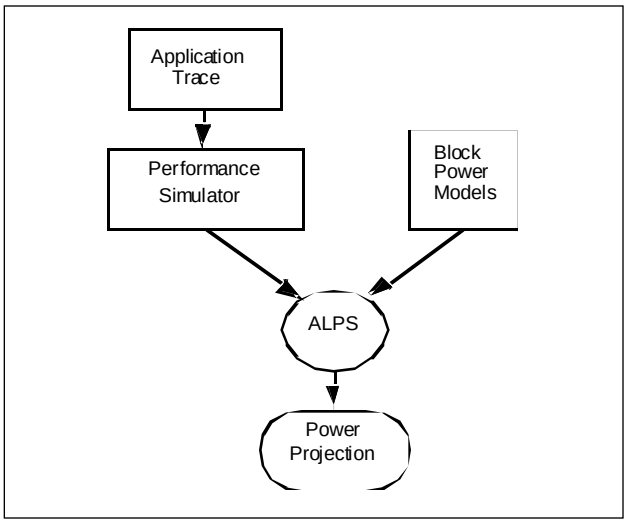

Figure 4: The ALPS flow

The ALPS methodology, flow charted in Figure 4, utilizes data from several sources. First of all, a high-level description of the design is used to identify which functional events are likely to have a significant impact on chip power consumption. This behavioral description is coupled with historical data on the power consumption of different types of functional blocks, as well as designer estimates, to approximate the energy cost of performing each of the functions associated with a given logic block. This results in a power model for each logic block of interest. We can then measure the frequency of occurrence for each of these events by utilizing the statistics generated by an architectural-level performance simulator for the design. The actual power estimate for a given instruction sequence is then made by ALPS as it executes the equation shown in Figure 5.

$$\text{Total Power} = \frac{\sum_{n=\text{Event}_1}^{\text{Event}_n} (\text{Energy}_n \leftrightarrow \text{Occurrences}_n)}{\text{Execution Time}}$$

Figure 5: Equation for total power

DEVELOPING THE BLOCK POWER MODELS

In order to focus model development efforts, it is necessary to understand the behavior of the processor to a level of detail sufficient to identify which functional events have the greatest impact on total power

consumption. By initially focusing only on those events that were expected to result in significant power consumption, we were able to quickly generate a simple power model, which allowed for incremental improvements as additional data became available. To clarify, it is known that each access to a large memory array (such as an on-chip cache) will cause the simultaneous transition of many address and data lines and have a noticeable impact on total power consumed. Conversely, the power associated with a small state machine that controls those cache accesses may have much less of an impact.

For similar reasons, events that are expected to occur very infrequently are also less important to model. As a further example, based on the frequency of occurrence, developing a power model for a logic block that generates a physical memory address for each load instruction is probably far more important than having a power model for a logic block that detects floating-point exceptions.

From a data collection standpoint, the functional-block-level information can be broken into two parts: information used to determine the activity level of a particular logic block and data to facilitate estimating the energy cost of each type of activity.

Each logic block in the design may perform one or more distinct logical functions. The activity level of the block as a whole is dependent on how often these functions are performed, the percentage of the logic that is associated with each function, and the length of time it takes to perform the function.

In the case of the Pentium 4 processor, we initially considered each unit to have five to eight key functions, and then added to this list, as additional design information became available. The initial list of activities was first based on a general understanding of the microarchitecture; it was later refined as the high-level processor definition code as the RTL for the block took form. Typically, extensive interaction with unit architects and designers was needed to clarify the events and activities key to each logic block.

POWER VALIDATION

Clock gating refers to activating the clocks in a logic block only when there is work to be done, and it is one of the key power-saving techniques employed on the Pentium 4 processor. When performing clock gating on a massive scale, two validation concerns arise. With overaggressive clock gating, logic failures can arise, where a block should have been awake, but either did not turn on quickly enough, or is turned off too soon. On the other hand, conservative clock gating will not disrupt correct functionality, but will result in wasted power. Functional checkers such as an architectural simulator do not report on such failures.

As one might anticipate, not all power-saving features have equivalent value. Therefore, a two-pronged approach was taken to finding these logic problems. One was fine-grained, carefully focusing on just the key power-saving features. The other was coarse-grained, examining all gated clocks for abnormal activity. For each approach, specific tools and methodologies were created to automate the process of finding these power-wasting logic problems. Use of these tools continued throughout the entire design process.

To uncover logical failures associated with clock gating, each Pentium 4 processor unit's Design Validation (DV) Test Plan was reviewed to ensure that the validator addressed the power down corner cases, and that such cases are included in the overall validation coverage figures.

THERMAL MONITOR OVERVIEW

Implementing a traditional thermal solution that accommodates the maximum power consumption of a leading-edge microprocessor like the Pentium 4 processor would have a significant impact on the system cost. To reduce this cost but retain a high level of reliability, an enhanced version of an existing mechanism used in the mobile computing segment, processor power modulation, was used in the Pentium 4 processor.

The processor power modulation mechanisms employed in mobile systems have taken two forms, both of which require the cooperation of external logic. The first mechanism involves slowly reducing the processor clock frequency, typically from its maximum supported frequency down to a lower frequency. The second mechanism involves the modulation of the processor STOPCLOCK signal (the pin named STPCLK#, while asserted, has the effect of stopping the clock to many internal elements of the processor). Since power consumption is proportional to operating frequency, both mechanisms have a similar effect on the power consumed by the processor.

The external logic that controls the power modulation of the processor could be activated by numerous stimuli, including detection of high processor or system temperatures, detection of low available battery power, or simply by a user selecting a low-power operating mode (with the goal of extending battery life).

In the case of a desktop computing system, one of the key constraints is the requirement to control the operating temperature of the processor. This requires being able to accurately measure the temperature of the processor silicon. Unfortunately, this is difficult to achieve with external temperature sensors. There is a significant delay between the time at which the processor silicon reaches a given temperature and the time at which an external temperature sensor notices the temperature change. Several solutions have previously been pursued, varying from attaching the temperature sensors to heat sinks, to

the processor package, or mounting it underneath the processor. Each solution has the same problem: reliable high-volume manufacturing is difficult.

More recently, portions of the thermal sensor (e.g., the thermal diode) have been located on the processor die. Even this approach has clear limitations. The temperature from one corner of the die to another can vary by a significant amount, so understanding sensor placement on the die is critical. Additionally, the rate at which the die temperature can change is increasing to the point that the currently available thermal sensor interface logic is too slow to allow reliable die temperature measurement or closed loop thermal control.

With the Pentium 4 processor, the objective was to enable accurate control of processor die temperature. The solution chosen integrates all portions of the power modulation mechanism including temperature detection through power control. When this feature is enabled, the processor is capable of operating with no further software intervention. In anticipation that software will eventually be required to control processor power dissipation, the architecture of the thermal monitor control logic has been created in such a way that it closely maps to the existing Advanced Configuration and Power Interface (ACPI) specification and software.

The thermal monitor architecture implemented on the Pentium 4 processor can be described as three separate but related functions: a mechanism for determining temperature, a mechanism for reducing the power consumption of the processor, and a means of controlling and providing visibility into each of these mechanisms. Each of these three functions is described in detail in the following sections.

TEMPERATURE DETECTION MECHANISM

All integrated circuits are designed to operate reliably within a defined temperature range. Outside of this range, there is no assurance that the integrated circuits will continue to function correctly. The die temperature at any given point in time is a function of the power consumed by the device (both at a given instant in time and in the relatively recent past), the collective thermal coefficient from the die through the heat sink, and the ambient environmental conditions.

The temperature at any given point on the die can be measured with the use of a diode and a precise current source. The voltage drop across a diode is dependent on both the current flowing through the diode and the temperature of the diode. By supplying a constant current, and measuring the voltage drop across a diode, we can get a reasonably accurate measurement of the temperature at which the diode is currently operating. By comparing this voltage to a reference point, we can determine when the temperature of the diode (and hence the portion of the die that contains the diode) is just below the maximum specified operating temperature. This is the only temperature with which we are concerned.

There are a couple of key factors that significantly impact the accuracy of such a thermal sensor. The characteristics of both the diode used as the thermal sensor and the transistors used to create the current source are dependent on the specific parameters of the manufacturing process. Many of the process parameters change slightly from one wafer to another or from one area on the wafer to another, affecting the temperature recorded by the thermal sensor.

The second factor impacting the accuracy of the thermal sensor is the fluctuation in the processor operating voltage (measured on the die rather than at the pins of the processor package). This noise can result in the thermal sensor comparator (which determines whether the die temperature has reached the maximum operating temperature) signaling that the die is too hot, when in fact it has not yet reached the critical temperature. Alternatively, this noise could also cause the comparator to incorrectly signal that the die is below the critical temperature.

The Pentium 4 processor implements mechanisms to account for both of these sources of error in the output of the thermal sensor. In the case of a microprocessor, the power consumed is a function of the application being executed. In a large design, different functional blocks will consume vastly different amounts of power, with the power consumption of each block also dependent on the workload. While the heat generated on a specific part of the die is dissipated to the surrounding silicon, as well as the package, the inefficiency of heat transfer in silicon and between the die and the package results in temperature gradients across the surface of the die. Therefore, while one area of the die may have a temperature well below the design point, another area of the die may exceed the maximum temperature at which the design will function reliably. Figure 6 is an example of a simulated temperature plot of the Pentium 4 processor.

As a result of the cross die temperature variations; it is very important that the temperature detection mechanism (the integrated thermal sensor in the case of the Pentium 4 processor) be located at the hottest spot on the die. As can be seen from Figure 6, there are clearly optimal locations for placement of the thermal sensor.

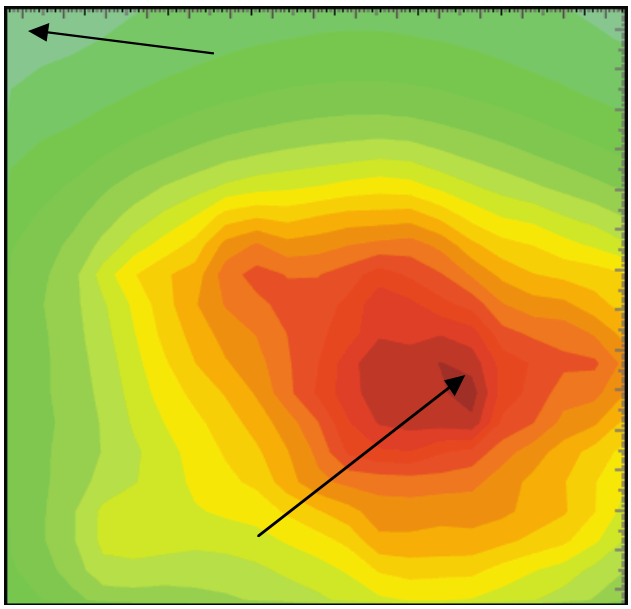

Figure 6: Simulated thermal plot of processor die

Because the hottest location on the die may change from application to application, it is important to confirm theoretical thermal maps of the die with actual measured data taken while various types of applications are executing. By evaluating the thermal maps for several classes of applications, it is possible to confirm the optimal location for the thermal sensor. Evaluation of these thermal maps also guides the selection of appropriate guard bands to be applied to the thermal sensor trip temperature. These guard bands are intended to ensure that as long as the temperature measured by the thermal sensor is below the maximum reliable operating temperature, there will be no location on the die that exceeds the maximum reliable operating temperature.

POWER REDUCTION MECHANISM

Once it has been determined that the die temperature is approaching the critical point, a mechanism is needed to quickly reduce power consumption, causing a drop in temperature. There are several key constraints in the design of this mechanism.

First, the latency between critical temperature detection and power reduction should be low. In this case, low latency refers to periods on the order of 100's of microseconds. Reaction times significantly longer than this would allow the die temperature to potentially reach a point at which it no longer operates reliably.

Second, the mechanism should be efficient. Here, efficiency refers to the ratio between power reduction and performance loss. An ideal mechanism results in a power vs. performance curve that is linear and crosses both axes at 0. In other words, if the power modulation mechanism results in a 10% performance loss while operating, it would also provide a 10% reduction in power consumption. Note an ideal relationship is only possible if frequency is the only variable.

Finally, the mechanism should add little or no cost to the design. Costs include those associated with die size, validation, platform impact, and risk.

After evaluation of a number of potential options, the Pentium 4 processor design team chose a mechanism that utilizes the existing architectural low-power logic (the StopClock architecture). The chosen mechanism essentially provides an internal STOPCLOCK request to the processor core.

This STOPCLOCK request results in the clock signal to the bulk of the processor logic being stopped for a short time period. While this clock signal is stopped, the power consumption of the processor is reduced to a small fraction of the maximum processor power consumption. Because the STOPCLOCK request is a relatively high priority interrupt, the delay between the request and the resulting power decrease is relatively short, typically much less than 1 microsecond.

In order to minimize any potential impact to the platform, the time period during which the clock is stopped is kept small. The design target limits the total time during which the processor is not executing useful code to a few microseconds. This includes both the time the clock is actually stopped and the overhead associated with stopping and restarting the clock signal.

THERMAL MONITOR CONTROL

The behavior of the power modulation mechanism can be controlled with an enable bit in a model-specific register. When enabled, the power modulation mechanism is automatically invoked whenever the thermal sensor indicates that the die is hot. The power modulation mechanism remains engaged until the die temperature drops below the critical value. While the default condition has this bit set to ”disabled”, it is required that the normal usage model would enable the thermal monitor functionality during the initialization process, and leave it enabled for as long as the system is powered on. This usage model provides the most robust processor thermal solution, as the processor can protect itself from most causes of overheating without any interaction by system hardware or software. The thermal monitor mechanism can also be invoked via the ACPI compatibility registers (see the section on ACPI interaction for details).

THERMAL MONITOR VISIBILITY

Although the thermal monitor mechanism implemented on the Pentium 4 processor can be configured to engage automatically and transparently, it may still be desirable to signal the thermal monitor state to the system hardware and operating system. In the Pentium 4 processor implementation, this signaling is provided via three means.

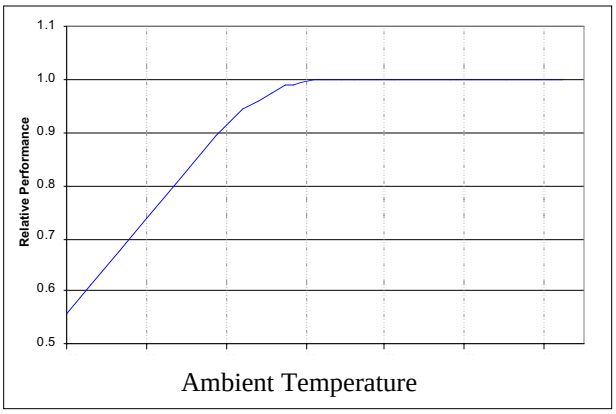

Figure 7: Example simulated system design curve

On the hardware side, an output signal reflects the state of the thermal sensor comparator. This signal is asserted while the thermal sensor indicates that the die temperature is at the maximum operating temperature, and is de-asserted while the temperature is below this point. This signal could be used by system hardware to take some action when the die temperature reaches the critical temperature. Note, however, if the closed loop thermal control is disabled, any cooling oriented action must be effective very quickly to prevent the processor from overheating and failing (see the section on the power-reduction mechanism). For example, using this signal to turn on an additional fan in the system would not be a sensible solution as the processor temperature could easily exceed its maximum operating temperature before the cooling effect of the fan is noticed.

On the software side, there are two, model-specific register-based, status bits of interest. The first reflects the state of the thermal sensor comparator. This information is identical to that provided by the output signal. The second bit is a "sticky" bit, which is set the first time the thermal sensor reaches the critical temperature, and it must be explicitly cleared by software or through a processor reset. An example use of the "sticky" bit would be for diagnostic software to determine if the processor has ever reached the critical temperature, which could be used as an indicator that the thermal solution performance has changed.

The final visibility mechanism consists of the ability to generate an interrupt request whenever there is a change in the output of the thermal sensor comparator. These interrupts can be generated in either direction; i.e., an interrupt can be generated when the thermal sensor output transitions from the "not hot" state to the "hot" state, and/or when the thermal sensor output transitions from the "hot" state to the "not hot" state. Each of these interrupts can be individually enabled or disabled.

PERFORMANCE

One of the primary themes behind the development of the described thermal monitor mechanism is the ability to reduce system thermal design costs without a perceivable impact on performance. Because the thermal monitor mechanism could impact performance if the processor reaches the critical temperature, it is valuable to understand how often, and for how long, the critical temperature could be reached while running real application code. These data allow system designers to design a solution that optimally balances system cost and the thermal performance required.

The performance impact resulting from the thermal monitor is dependent on both the processor power consumption and the thermal solution. By generating a curve of thermal monitor performance impact vs. system thermal capability, the system designers can determine the design point that is optimal for their target market.

During the development process of the Pentium 4 processor, the Architectural-Level Power Simulator (ALPS) was used to project the power consumption of a range of applications. By using the power-consumption projections of the ALPS, along with the expected thermal characteristics of the Pentium 4 processor package, it was possible to project the temperature of the Pentium 4 processor die at a given point in time while running the applications of interest.

The resulting temperature vs. time data could then be used to project when the processor would reach the critical temperature. Using the characteristics of the thermal monitor mechanism, along with the package characteristics, it is possible to project how long the thermal monitor mechanism would remain active. The process described was automated and was used to generate curves of processor performance vs. system thermal design capability. Figure 7 shows an example of one of these curves. As can be seen from Figure 7, the thermal monitor mechanism implemented has the potential for significantly reducing the system thermal design point, without perceivably impacting processor performance.

INTERACTION WITH THE ADVANCED CONFIGURATION AND POWER INTERFACE SPECIFICATION

The Advanced Configuration and Power Interface (ACPI) specification defines a hardware and software environment that allows operating system software complete visibility and control of system configuration and power management. The specification describes a set of valid processor operating states and the allowable transitions between them. The upper four states defined for the processor are as follows:

1. C0—normal operation
2. C1—a low-power, low-latency state that assumes no support from chipset logic that retains all cached context

3. C2—a lower-power, slightly longer latency state than C1 that requires chipset support but still retains cached context
4. C3—a still lower power, longer latency state that also requires chipset support but one in which the cached context may be lost

Systems based on the IA-32 architecture will typically map the use of the HALT (HLT) instruction to the C1 state, the STOPCLOCK assertion to C2, and Deep Sleep (removal of the processor clock input signal) operation to the C3 state.

A documented sub-mode of the ACPI, C0 state is known as Clock Throttling (the thermal control functionality on the Pentium 4 processor would map to this sub-mode of the ACPI spec). In this mode, the operating system accesses logic to assert the STOPCLOCK signal with some predetermined duty cycle prior to the Pentium 4 processor, this logic had been resident in the chipset). The term "duty cycle" is used to refer to the characteristics of the signal applied by the chipset to the processor's STOPCLOCK pin in order to reduce processor power dissipation.

The register that is defined to enable and configure Clock Throttling is named the Processor Control register (P_CNT) by the ACPI specification. This 32-bit register has bits defined to both set the Clock Throttle (power control) duty cycle and force the thermal control to begin. The actual width and offset within P_CNT of the duty cycle field can be configured by a system developer. The Pentium 4 processor has implemented this P_CNT register in internal Model Specific Register (MSR) space. The three duty cycle bits implemented in the Pentium 4 processor's control register give software the ability to define seven levels of power control, with one value (0) reserved.

The incorporation of the P_CNT register in the processor provides the operating system with the ability to perform thermal control on a per processor basis even when there are multiple processors in a system. The Pentium 4 processor does not support Multi-Processor (MP) system configurations; however, there will be future MP capable IA-32 processors based on this same microarchitecture targeted at the server and workstation market. It should be noted that, to date, chipsets have only implemented a single set of ACPI Clock-Throttling registers, and that chipsets have a single STOPCLOCK pin. The net impact is that a per processor Clock Throttle solution does not currently exist for MP systems. Incorporation of a P_CNT register into the processor solves this issue without requiring the addition of multiple STOPCLOCK pins in the chipset.

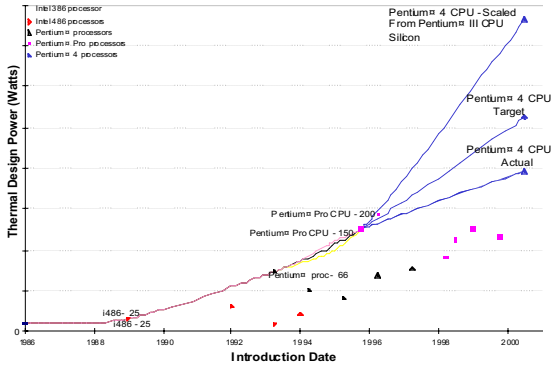

Figure 8: Pentium® 4 processor power with historical trends

It is interesting to observe that the Thermal Monitor and Clock Throttle functions, although similar and intended to cooperate/inter-operate, are designed to allow control of fundamentally different system parameters. The Thermal Monitor is intended to very closely control the processor die temperature, ensuring that the processor temperature remains within the specified range. The Clock Throttling defined by ACPI is intended to allow the operating system to modulate the processors' power dissipation in order to control the ambient temperatures that may impact other components within the system.

CONCLUSION

At the start of the Pentium 4 processor project, the design team formally committed themselves to lowering the processor power consumption by 20% from initial power projections . The team also committed to lower the thermal design point of the system by 40% without perceivably impacting application performance, while maintaining processor reliability. These commitments were met in the initial version of the Pentium 4 processor. For reference, Figure 8 depicts the level of power savings that the Pentium 4 processor achieved, superimposed on historical thermal design power data.

The power reduction achieved resulted largely from the extensive application of clock gating and unit power-down techniques. The addition of the thermal monitor feature enables what is essentially a processor that is capable of managing its own thermal profile to operate efficiently within almost all thermal environments.

ACKNOWLEDGMENTS

The authors thank the Pentium 4 processor design team for their significant effort to minimize the power consumption of the processor, with special thanks to Bindu Lalitha, Kimberly Weier, Bret Toll, Kan Chu, and Linda Rankin.

AUTHORS' BIOGRAPHIES

Stephen H. Gunther joined Intel in 1992. He has spent much of the last eight years in the Desktop Processor Group working on techniques for reducing the power consumption of Intel microprocessors. Steve received his B.S. degree in Computer Engineering from Oregon State University in 1987. He is currently involved in the development of future microprocessors. His e-mail is steve.h.gunther@intel.com

Frank Binns obtained a B.S. degree in electrical engineering from Salford University, England. He joined Intel in 1984 after holding research-engineering positions with Marconi Research Laboratories and the Diamond Trading Company Research Laboratory both of the U.K. Frank has spent the last 16 years with Intel, initially holding technical management positions in the Development Tool, Multibus Systems and PC Systems divisions. Frank's last eight years have been spent in the Desktop Processor Group in Technical Marketing and Processor Architecture roles. His e-mail is frank.binns@intel.com

Douglas M. Carmean is a principal architect with Intel's Desktop Products Group in Oregon. Doug was one of the key architects, responsible for definition of the Intel® Pentium® 4 processor. He has been with Intel for 12 years, working on IA-32 processors from the 80486 through the generation beyond the Intel Pentium 4 processor. Prior to joining Intel, Doug worked at ROSS Technology, Sun Microsystems, Cypress Semiconductor and Lattice Semiconductor. Doug enjoys fast cars and scary, Italian motorcycles. His e-mail address is douglas.m.carmean@intel.com

Jonathan C. Hall graduated from Pennsylvania State University with a B.S. degree in Computer Engineering in 1993. He completed his Master's degree in computer science at Rice University in 1996. For the last four years he has worked for the IA-32 architecture and performance team in the Desktop Processor Group at Intel. Jonathan is interested in processor power, performance, and compilation. His e-mail is jonathan.c.hall@intel.com