Potential Thermal Security Risks

Puyan Dadvar and Kevin Skadron
Department of Computer Science
University of Virginia
Charlottesville, VA 22904
{puyan,skadron}@cs.virginia.edu

Abstract

Hardware and software techniques for controlling a microprocessor's power and cooling have the undesirable side effect of creating a security risk. They allow a malicious program to control the chip's operating temperature and potentially cause denial of service or even permanent damage. This paper provides an overview of the various vulnerabilities, their costs, and offers preliminary suggestions on how to reduce these risks.

Keywords

Thermal, temperature, power, cooling, security, virus, failsafe, overheating, throttling, overclocking, DVS

1. Introduction

A chip's power dissipation and hence heat dissipation are program-dependent. This means that malicious programs can potentially be written to manipulate the way a computer chip dissipates power, intentionally affecting battery life and operating temperature. The risks associated with such "power exploits" are poorly understood. This paper focuses on thermal behavior and describes possible types of thermal attack, the types of damage that are possible, which attacks require supervisor privileges, and offers preliminary suggestions on how designs can be changed to prevent or mitigate these attacks. To the best of our knowledge, this is the first work to consider thermal attacks and show that serious damage is possible. We are not aware of any actual attacks of this form, but show that such attacks are possible, may have severe consequences, and require study by the research and design communities.

An unfortunate corollary of the exponential increase in performance associated with Moore's Law for high-performance chips is an exponential increase in power density and hence cooling requirements. Even for low-power applications like mobile and embedded systems, where form-factor constraints mean that expensive cooling may not even be feasible, demand for higher performance is pushing the limits of these systems' cooling capabilities. Even in many high-performance systems, cooling costs are severely constrained. For example, even in a typical desktop computer, the cooling solution may only be allocated a few dollars! For a given price point, chips that need more expensive cooling solutions reduce the profit margin.

One solution that allows high performance while mitigating cooling costs is to design the cooling system for less than the worst case. Most reasonable workloads do not induce worst-case power dissipation, let alone over a sustained period of time. This means that designing the cooling for worst-case behavior is actually costly over-engineering. The cooling solution can instead be designed for the worst *typical* program, thus reclaiming excess design margin. Then, to guard against exceptional cases in which high power dissipation is sustained long enough to exceed the capacity of the lower-cost cooling solution, high temperature-sensor readings trigger the chip itself to autonomously alter its behavior and reduce its heat dissipation, for example by operating at a lower voltage and frequency. This *dynamic thermal management* (DTM) trades off fixed packaging costs for workload-dependent performance cost. A number of high-performance CPUs already use some form of DTM, for example the Intel Pentium 4 and Pentium M, the Transmeta Crusoe and Efficeon, the IBM Power5, and the AMD Athlon. A great deal of recent research has explored how to design DTM solutions to minimize this performance cost, e.g.. [1][2][3][4]

Unfortunately, the exact attribute that makes DTM appealing for normal workloads creates a security vulnerability. A malicious program can force DTM to engage and slow the system down. And if the DTM mechanism can be controlled by software, even more serious consequences are possible, potentially even destruction of the CPU. Software control of other features that affect operating temperature, like fan speed and chip voltage, only exacerbate this problem.

In this early work, our goal is to bring this threat to the attention of the research community. We first provide a brief background on the thermal-management technologies that create these security risks. We then show qualitatively how these capabilities can be used to cause undesirable behavior and some potential consequences. We cannot yet provide detailed, quantitative results, but intend our early work to illustrate the potential security risk. Finally, we offer some preliminary suggestions on how designers can prevent or at least mitigate such security risks, and list some areas for future work. We argue that both hardware and software solutions are needed.

2. Background

2.1. Thermal Monitoring and Control

To illustrate the detection and management of thermal stress, this section gives an overview of thermal management in the Intel Pentium 4, a thermal-management implementation that we regard as generally robust, although it is still vulnerable to some undesirable behavior caused by ill-behaved or malicious software.

The Pentium 4 employs two on-chip sensors. Details can be found in Intel documentation for various versions of the Pentium 4, e.g. [5][6][7][8]. The chip's internal thermal

control circuit uses an internal thermal diode and compares it to a reference current. Readings from this sensor are not externally visible. This sensor is placed near the portion of the CPU that is expected to be the hottest under normal operation. It is not clear if this location is guaranteed to always represent the hottest spot, or whether it is possible under unusual workloads for other locations on the chip to become hotter

When the temperature exceeds a factory-preset temperature that indicates thermal stress, the thermal control circuit begins throttling CPU activity by stopping the clock (and hence stopping all activity on the processor) for a short period of time, e.g. two microseconds. After the clock is re-enabled, if temperature remains high, the clock will be stopped again, and this process continues with a software controlled duty cycle. The exact duty cycle is a factory-preset value, but is typically close to 50% (i.e., the clock is enabled for 2 μs, then disabled for 2 μs). This "automatic" mode duty cycle cannot be changed, but throttling can be disabled altogether through internal configuration registers on the CPU.

Note that the operating system can implement its own thermal management policy, e.g. in conjunction with ACPI. An operating system that supports ACPI can implement a thermal policy that will manage CPU temperature in three different ways. The first policy is active cooling such as turning on a fan. The second policy is passive cooling which reduces power consumption of the CPU to reduce temperature. An example of passive cooling is throttling the processor clock. Lastly, there is a critical trip point at which the OS conducts a graceful shutdown. Each policy has temperature thresholds that can be set by the OS to inform ACPI of when to engage or disengage that policy. [9] These thresholds can be lower than the CPU's internal temperature thresholds.

The thermal control circuit can be engaged "on demand" by the operating system to throttle the CPU using a duty cycle chosen from a range of 12.5%-87.5%. This might be used by the operating system as part of ACPI thermal management or for other purposes.

When the processor engages throttling, it also asserts an external pin (PROCHOT) to inform the operating system, in case this may influence its scheduling or power/thermal management. For systems which have been designed with more robust and expensive cooling solutions, where DTM should never engage, PROCHOT may also indicate cooling failure or a radically misbehaving workload.

A second, failsafe thermal control circuit engages at a second, factory-preset temperature, approximately 135°C, when immediate damage in imminent, and automatically resets the chip. This shuts down the processor, asserts an external pin (THERMTRIP), and forces the system to be rebooted. Unlike thermal throttling, this mechanism cannot, to our knowledge, be disabled. The failsafe is needed in case the DTM response that is designed for minimal performance overhead cannot control the temperature, or in cases where the cooling solution fails (for example, a fan failure or a detached heat sink.)

The threshold temperatures for the thermal control circuit and the preset duty cycle are configured on a part by part basis in conjunction with nominal voltage and frequency ratings in order to account for manufacturing variations and to obtain optimal performance and energy efficiency. The threshold temperatures for a specific chip cannot be determined to our knowledge, because the temperature difference between the software-visible diode and the internal sensor that controls throttling is unknown. The throttling duty cycle can be determined empirically, e.g. with a multimeter.

The Pentium 4's second on-chip sensor is a thermal diode that is software visible. The thermal diode is not located near the thermal control circuit's temperature sensor, and hence not located near expected hot spots. Intel documentation explicitly warns that this second diode's temperature readings are not well correlated with temperatures in the hot spots. Nevertheless, this is the sensor that is externally visible. The diode produces a voltage across two external pins that can be converted using external A/D hardware on the motherboard. ACPI uses this reading to implement its thermal management policy. In our test system, this external diode gives a reading of approximately 67°C when automatic thermal throttling begins to engage. This probably corresponds to $T_j$max values of approximately 80-100°C. We find that throttling engages gradually, which matches the Intel documentation's observation that the thermal control circuit uses hysteresis to avoid unnecessary oscillation.

We have not been able to find any documentation giving exact temperatures, and the lack of any other software visible sensors makes it difficult to determine actual values for the on-chip temperature gradients. Unfortunately, we do not have access to infrared or thermometry equipment that might allow these kinds of measurements.

2.2. Fan Control

Many motherboards dynamically control fan speed in order to conserve energy and minimize noise. (Many fans can be uncomfortably loud at full speed.) A database listing motherboards that allow fan speed control is available at [10]. This means that software, through the fan controller's configuration registers, can vary the voltage to the fan or even disable it.

2.3. Frequency and Voltage Scaling

Dynamic voltage and frequency scaling (DVS) allow the CPU to operate in a lower-performance but also lower-power state when the load is light, and scale up to higher-performance states only when running CPU intensive programs. This technique takes advantage of the fact that power is roughly proportional to V^2f, so that power savings is roughly cubic with respect to the loss in performance (i.e., frequency). Indeed, many chip manufacturers brand their DVS technique for marketing purposes. Examples include various forms of Intel's SpeedStep and AMD's PowerNow.

To manage the DVS setting in response to changing workload conditions, the OS typically has the ability to control voltage and frequency. An interesting side effect of this capability has been its use for "overclocking," setting the frequency and possibly the voltage to higher values than the

CPU is rated for. Recently, the option to configure the overclocking via software in Windows has become an option. This software makes it especially convenient to vary the CPU and memory-bus frequency, which is problematic from a security standpoint.

The interface to the DVS for Pentium 4 and AMD is made available by writing to internal CPU configuration registers. With out test system, the AMD DVS interface does not allow the OS to overvolt or overclock a CPU (but using an unreasonably low voltage is possible). However, some motherboard manufacturers make software available to allow overclocking. The process they use to overclock the CPU we have not been able to determine, however, we speculate it is done through an on-board embedded controller that sets the front-side bus frequency and voltage, which in turn feeds through the CPU's clock multiplier to produce overclocking.

With SpeedStep and PowerNow, voltage and frequency pairs are set through internal CPU configuration registers. These internal configuration registers are accessed through device drivers such as the module msr.o on Linux. On our AMD 1800+, nominally rated at 1.55 V and 1.5 GHz, we are able to set the frequency between 550 Mhz and 1.5 GHz with voltages between 1.10 V and 1.55 V in .05 V increments. With the Asus motherboard overclocking technology, settings are altered through a windows application known as AIBooster. This software allows us to set our 1.52 V, 2.6 GHz P4 to frequencies from 1.3 GHz to 3.3 GHz. Voltages can be varied from 1.1 V to 1.95 V in .0125 V increments. Note that neither processor enforces specific voltage-frequency pairings.

2.4. Hardware Access Privileges

To directly manipulate many of these power- and thermal-management technologies and accomplish the attacks we describe, a malicious program must typically obtain operating-system privileges, in order to access memory-mapped I/O ports. privileged CPU registers, and the fan control and overclocking facilities supported or even directly provided by many motherboard manufacturers.

Obtaining privileges to access these interfaces can be accomplished through a variety of attacks that have been well-documented in the security community and popular press, like weak passwords, Trojan Horses, email viruses, buffer-overflow (aka "stack-smashing") exploits, device drivers with relaxed permissions that expose some of this functionality to user-level programs, or simple user gullibility. A further vulnerability is that many Windows machines are still used for normal purposes in "Administrator" mode, which possesses full privileges—a very risky practice. Final

Thermal throttling on the P4 is controlled through internal configuration registers. These CPU registers require any code that accesses them to be in kernel mode. In order to put code into kernel mode one requires administrator or "root" privileges.

3. Results

3.1. General Specification of Hardware and Software Used in Our Experiments

Our Pentium 4 experiments were performed with an Asus P4P800 motherboard, a 1.52 V, 2.6 GHz "Northwood" CPU, Windows XP Service Pack 1, a Winbond W83627THF sensor chip, SpeedFan fan-control software, and Asus AiBooster DVS control. "Northwood" is the code name of a particular generation of the Pentium 4; the most recent Pentium 4s are the "Prescott" generation. We primarily focus on the Intel system in this paper due to greater experience with this platform. We used the CPU MSR and CPU RightMark programs to monitor the throttling status and verify that automatic rather than on-demand mode was in use. We also conducted some experiments with a Compaq Evo N1015V laptop with an AMD 1800+, 1.55V, 1.5 GHz AthlonXP CPU, Debian Linux Kernel 2.6.6, and the "xmbmon" fan-control software.

The Winbond sensor chip is a PC health monitoring chip. It can provide the current CPU temperature and can control fan speeds. The Winbond chip also has the capability of informing the system in case the CPU runs the risk of damaging itself due to high temperatures. The fan control and temperature readings are accessed through the sensor chip's internal registers. [11]

There are several programs that can be used in order to place the CPU under thermal stress. They maximize instruction throughput to maximize CPU activity. The programs that we used to cause stress on our system are RightMark, CPUBurn, and Prime95, all of which are popular in the overclocking community for testing the stability and reliability of overclocked configurations. More information regarding these programs can be found on websites for overclocking enthusiasts. We will refer to these types of programs generally as "thermal stressmarks."

3.2. Thermal Security Risks Observed

We were able to demonstrate several scenarios in which the capabilities described in Section 2 were used to create denial-of-service situations, data-integrity problems, or the possibility of permanent damage to the CPU. These results only present our early, qualitative findings and are not an exhaustive listing of thermal security risks. Our goal is to show the importance of the thermal security risk. All the following results were achieved using software mechanisms alone, without modifying the hardware.

Denial of Service – Thermal Throttling. We were not able to cause thermal throttling to engage under normal operating conditions at room temperature, even with a stressmark. We were able to cause thermal throttling to engage by partially blocking the system's air vents, and it is possible that even without blocking the vents, throttling might engage under more extreme environmental conditions.

Of greater relevance from a security standpoint, we were able to cause thermal throttling to engage for the Intel Pentium 4 by using software to disable the fans and then running a thermal stressmark. In the default throttling configuration, this reduces performance by approximately

50% in our system, effectively a denial-of-service attack. In a warmer environment, or with a higher voltage, throttling would presumably engage more quickly, and temperatures could possibly even be raised high enough to engage the failsafe reset mechanism.

The newer, multi-threaded Prescott cores are reported to have a throttling trigger temperature very close to the expected operating temperature under normal workloads. [12] So throttling could be a serious problem for processors that run within specification but very close to the trip temperature—even for legitimate workloads in normal environmental conditions with the fan operating properly. For example, we speculate that even in a climate-controlled environment, an attack that launches two concurrent stressmark threads may be enough to trigger throttling Finally, malicious code could simply engage throttling directly, using the on-demand feature.

Denial of Service – Reset. Whether or not throttling is disabled, a thermal stressmark can potentially raise the temperature high enough to engage the Pentium 4 failsafe and reset the computer. At best this is a nuisance, and at worst can present a sustained denial-of-service attack. If the computer has been configured (by a virus, for example) to run the same stressmark on reboot, the computer will repeatedly reset. We accomplished this scenario by turning off the fans. We were able to accomplish reset from user level by taking advantage of a popular fan-control device driver that creates relaxed permissions when installed by administrator. With throttling disabled, the approximate time for reset decreased from 40 minutes to 5 minutes. We also observed that resets (because the computer shuts down abruptly) occasionally corrupted the journaling file system, requiring the file-system self-repair procedure. We did not attempt this same test on the Athlon but a similar risk is present.

Denial of Service and Data Integrity – DVS. Although not a thermal risk per se, we note that the ability to control the DVS setting from software means that a malicious program can simply reduce the voltage or frequency. Reducing frequency reduces performance directly. And if voltage is reduced far enough below the CPU's rating for a given frequency, timing errors can occur in the circuitry, sometimes leading to program crashes or system reboots, but sometimes leading to latent arithmetic errors which may not always be evident to the user. Furthermore, as temperature rises, the voltage does not need to be as low to cause errors. For example, rounding errors were observed at 1.28V, 2.6 GHz, and 76°C, compared to 1.23V, 2.6 GHz, and 31°C. Data integrity problems occurred on our P4 when we reduced the voltage from the typical ~1.57 volts to ~1.15 volts at the nominal maximum frequency of 2.6 GHz. An increase in the frequency aggravates the problem. We accomplished these voltage and frequency changes through the Asus AIBooster overclocking program, which allows one to overvolt and undervolt. With the AMD AthlonXP we observed internal compiler errors (data errors causing a compiler to abort), system freezes, and program instabilities/crashes in which various programs failed. This was accomplished through the PowerNow interface, which only allows undervolting, not overvaulting.

Raising the voltage above the specified maximum, especially in conjunction with higher frequency, will obviously accelerate the onset of thermal stress.

Gradual Damage – Accelerated Aging. Reduction of lifetime under high temperature and voltage conditions is possible. Our Pentium 4 can be operated at a sustained temperature of 95°C on the externally visible thermal diode. This was accomplished using thermal stressmarks by shutting off the fan, and leaving the thermal control circuit on to throttle and attempt to maintain the temperature. 95°C may not seem high, but is 20° above the observed maximum operating temperature that the thermal control circuit enforces. [5] Keep in mind that the temperature sensor on the Pentium 4 is not co-located with the hottest part of the chip, so the actual hot spot on the chip may be substantially hotter than the temperature sensor's reading. With AiBooster we were also able to set the voltage to 1.98 volts from the typical 1.63 V, which will exacerbate the stress, because many aging mechanisms are dependent both on temperature and voltage.

Excess temperatures will accelerate a number of IC failure mechanisms, like electromigration and gate-oxide breakdown. Rapid cycling between hot and cold temperatures can also induce thermo-mechanical stress, possibly damaging the chip, package, or both.

Permanent Damage – Disabling the Failsafe. The Intel failsafe cannot be disabled as far as we know, but older motherboards for AMD AthlonXP processors did not provide an adequate failsafe. [13] Although that problem was quickly rectified, even newer motherboards may allow the failsafe to be bypassed by disabling the thermal-emergency shutdown procedure in the BIOS. Obviously, we did not try this, but the same scenario that engaged the failsafe with the Pentium 4 would overheat and quickly destroy the processor. This is not reported to criticize AMD systems, but simply to show the importance of a failsafe mechanism that is hardwired. AMD has built a thermal trip mechanism in its latest generation of processors: the Athlon64. Similar to the P4, once the failsafe is engaged the CPU will shutdown its internal clock and inform the motherboard to reset through a THERMTRIP pin [14].

4. Discussion and Recommendations

The appeal of a thermal attack is its novelty and hence the likelihood that this type of attack may be difficult to pinpoint and stop even if the user realizes something is wrong with the system. An additional appeal is the possibility of dramatic destruction of the CPU. A popular video of a heat sink failure shows the CPU literally burning up in smoke. [13] Although we are not aware of any thermal attacks to date, we expect that it is only a matter of time before crackers attempt to deploy a thermal virus.

Despite continuing work on computer security, the risk of compromising the operating system remains substantial, as witnessed by continuing, frequent break-ins—this despite continual security patches from major operating system vendors. CPU and motherboard manufacturers must develop

hardware solutions to prevent those thermal attacks that are possible with a compromised OS.

Hard-wired failsafes that cannot be disabled are a must. Fan noise is a growing problem (Asus QFan and AMD Cool 'n Quiet are examples of the attention large companies are giving to this consumer concern), but fan control must be managed by hardware to provide acceptable noise control while eliminating the need for fan-control programs or OS intervention. At the very least, thermal management hardware must be able to override dangerous software-specified fan settings.

Voltage and frequency settings should be restricted by hardware to match the safe operating ranges of the CPU. Intel and AMD have attempted to lock the maximum voltage and frequency at which their systems can operate, but motherboard manufacturers, perhaps in a bid to appeal to overclocking enthusiasts, have to some degree circumvented this. According to experiments performed by Anandtech, Intel has an overclock lock on its 925X model chipset that prevents a user from overclocking their processor 10% above its intended clock rate. If a user attempts to overclock above 10%, then the system will just reboot or shutdown. However, Asus was able to circumvent this lock and allow the CPU to be increased to 25% over the intended clock rate. [15]

Stricter limits may need to be enforced, and at the very least, thermal management must be able to override overclocked voltage and frequency settings. Limits on the minimum voltage for each possible frequency are also needed.

Throttling can potentially be engaged from the user level by a thermal stressmark. Although we have not been able to test a Pentium 4 Prescott core, the combination of higher leakage currents, higher operating temperatures, and multi-threading make it possible that a thermal stressmark will engage throttling even in normal environmental conditions. Here operating system solutions play an important role, by providing techniques and policies to enforce not just fair scheduling, but also "fair heating" that ensures all programs get a fair share of full-speed CPU time, even if this means limiting execution of a "hot" program that engages throttling. This requires the operating system community to develop techniques for attributing heating to the responsible programs.

Chips may also need more robust on-chip temperature-sensing organization than provided in most current systems, with multiple interconnected sensors. Little data is available about possible on-chip temperature gradients, but we are concerned that with just one or two on-chip sensors, a carefully designed thermal attack could heat up a unit far enough from the sensor that thermal damage would result without triggering thermal protection.

These are just a few suggestions on important areas for future research and some techniques that can help mitigate the security risk.

5. Conclusions and Future Work

The ability for a program to generate thermal stress creates a security risk not only for high-performance system, but for any system that operates near the limits of its thermal packaging. Any system that uses dynamic thermal management, and any system that exposes aspects of its cooling system to operating system control, is vulnerable. Unfortunately, for both reasons of costs and form factor, DTM seems unavoidable for a growing class of systems.

A comprehensive study of thermal and power security vulnerabilities is needed. Research is needed to identify attributes of thermal attacks that might allow them to be identified by an intrusion-detection system, to understand how other chip-level power-management techniques may present security vulnerabilities, and to identify both hardware and operating-system techniques for thermal management that are less likely to affect legitimate programs. This paper has focused on thermal vulnerabilities in personal computers; research is also needed to evaluate not just thermal but also energy vulnerabilities in other classes of systems like servers and various mobile and embedded systems, as well as other system components like disk drives, memory chips, graphics acceleration cards, and other peripherals or accessories. Runtime thermal management presents additional challenges in real-time systems, and techniques are needed to reconcile these apparently contradictory features.

The advent of multiple processors on a single chip, in the form of high-performance chip multiprocessors as well as a variety of systems-on-a-chip, add further interesting dimensions to the problem, both in terms of possible risks (each component may present different vulnerabilities) and possible thermal management solutions (e.g., new scheduling strategies).

In the meantime, thermal engineers and CPU, motherboard, and operating-system designers must cooperate to limit or eliminate the risks described above. Robust thermal solutions should assume that the OS has been compromised.

Acknowledgments

This research has been funded by grant no. W911NF-04-1-0288 from the U.S. Army Research Office and two Research Experience for Undergraduate supplements on NSF grant nos. CCR-0105626 and CCR-0133634. We would also like to thank the anonymous reviewers, Anita Jones, Jesse Foster, and Kyeong-Jae Lee for their helpful comments and suggestions.

References

1. D. Brooks and M. Martonosi, "Dynamic Thermal Management for High-Performance Micro-processors", Proc. 7th Int'l Symp. High-Performance Computer Architecture, IEEE CS Press, pp. 171-182, 2001.
2. K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. "Temperature-Aware Micro-architecture", Proc. 30th Ann. Int'l Symp. Computer Architecture, IEEE CS Press, pp. 2-13, 2003.

3. J. Srinivasan and S. Adve, “Predictive Dynamic Thermal Management for Multimedia Applications”, Proc. 17th Int’l Conf. Supercomputing, ACM Press, pp. 109-120, 2003.
4. J. Donald and M. Martonosi, “Temperature-Aware Design Issues for SMT and CMP Architectures”, Proc. 2004 Workshop on Complexity-Effective Design, 2004.
5. Intel Corp., “Intel Pentium 4 Processor with 512-KB L2 Cache on 0.13 Micron Process and Intel Pentium 4 Processor Extreme Edition Supporting Hyper-Threading Technology: Datasheet”, Order number 298643-012, 2004.
6. Intel Corp., “Mobile Intel Pentium 4 Processor-M: Datasheet”, Order Number 250686-007, 2003.
7. Intel Corp., “Intel Pentium 4 Processor with 512-KB L2 Cache on 0.13 Micron Process Thermal Design Guidelines: Datasheet”, Order Number 252161-001, 2002.
8. Intel Corp., “IA-32 Intel Architecture Software Developer’s Manual. Volume 3: System Programming Guide”, Order number 253668, 2004.
9. ACPI4Linux Documentation, http://acpi.sourceforge.net/documentation/thermal.html
10. A. Comparetti, “SpeedFan’s Supported Hardware”, http://www.almico.com/forumindex.php
11. “Winbond LPC I/O W83627THF”, Winbond Official Technical Documentation, http://www.winbond.com.tw/c-winbondhtm/partner/PDFresult.asp?Pname=925
12. S. Garmatyuk, “Testing Thermal Throttling in Pentium 4 CPUs”, http://www.digit-life.com/articles2/p4-throttling/
13. T. Pabst and F. Völkel, “Hot Spot: How Modern Processors Cope With Heat Emergencies”, Tom’s Hardware, http://www.tomshardware.com/cpu/20010917/
14. “BIOS and Kernel Developer's Guide for AMD Athlon 64 And AMD Opteron Processors”, AMD Official Technical Documentation, http://www.amd.com/us-en/assets/content_type/white_papers_and_tech_docs/26094.PDF
15. W. Fink, “Intel 925X: Exploring the Overclock Lock”, Anandtech, http://www.anandtech.com/cpuchipsets/showdoc.html?i=2092