Performance loss during normal operation in a Dell Latitude E6500 laptop due to processor and bus clock throttling

Randall Cotton - recotton@earthlink.net
Revision 1.1 - July 31, 2009

Introduction:

This article describes how processor performance can decline by more than 95% in a Dell Latitude E6500 running Windows XP Professional (Service Pack 3) by running routine software at normal operating temperatures due to activation of the following clock throttling mechanisms present in Intel Core 2 Duo processors:

I. Performance State Transitions
II. Software-controlled Clock Modulation (also called On-Demand Clock Modulation)

These clock throttling features are designed to be invoked to conserve energy when a processor is idle or to head off an overheating condition. However, in the tested Dell Latitude E6500, they are engaged at normal operating temperatures even when demand for processing power is high. This can result in severe loss of performance under routine use at normal ambient room temperatures. This loss of performance can remain until the system is rebooted (and, since it is temperature-dependent, the performance loss can conceivably remain across reboots).

To illustrate and reliably reproduce this phenomenon, various methods are presented that raise internal system operating temperatures enough to cause the throttling mechanisms to be invoked while monitoring key parameters such as processor temperatures and operating frequencies. In some cases, freely available third-party processor-stressing software is used, but in other cases, even simple Windows-native processes like Calculator and a 4-line looped batch file that outputs scrolling lines of text are enough to trigger substantial loss of processor performance. In the worst case, all of the following takes place:

I. Reduction of internal CPU clock frequency from 2261 MHz to 798 MHz
II. Reduction of FSB (Front Side Bus) clock frequency from 266MHz to 133MHz
III. In addition to the above, overall reduction in the CPU effective clock rate by a factor of eight due to Software-controlled Clock Modulation.

note: the term CPU stands for Central Processing Unit, referring to a PC's main processor chip
note: the term Front Side Bus is the technical name for the main data path between the CPU and the rest of the system

Even setting aside the negative performance effect of FSB downshifting in II above, the effective processing power is reduced to 1/8 of 798 Mhz = 100 MHz. This is a reduction to less than 5% of full capacity (100/2261 = 4.42%).

It is noteworthy that all performance loss takes place with neither notice nor explanation to the user.

Background: Processor Clock-throttling Mechanisms

The Dell Latitude E6500 features an Intel Core 2 Duo processor, which offers several mechanisms that can be used (typically by operating systems such as Microsoft Windows) for both thermal management and power conservation. Background information follows on the two mechanisms already mentioned.

A. Performance State Transitions

Many of Intel's CPUs are designed to run at a variety of performance states. A performance state, or P-state, is generally a combination of processor frequency (such as 2.26GHz) and a voltage level (such as 1.2V) [1]. Lower frequencies provide less processing performance, but require lower voltages supplied to the processor and therefore consume less electrical power. Typically, Intel CPU's in PCs use 4 performance states. There can be a wide range of processor performance among these states. In the Dell Latitude E6500 tested for this article under Windows XP, the following states are used by default:

Performance State	FSB Clock Frequency	CPU Clock Frequency	Est. CPU Voltage
P0	266 MHz	2261 MHz (8.5 x FSB frequency) ***with*** Intel Dynamic Acceleration	1.2V
P1	266 MHz	2261 MHz (8.5 x FSB frequency) ***without*** Intel Dynamic Acceleration	1.1375V
P2	266 MHz	1596 MHz (6 x FSB frequency)	1.0V
P3	133 MHz	798 MHz (6 x FSB frequency)	0.925V

Note: The terminology P0, P1, etc., referring to P-state 0, P-state 1, etc., is from the ACPI standard – Advanced Configuration and Power Interface. That standard and its relevance are described later in this article.

Note: Intel Dynamic Acceleration refers to a feature whereby one of the two processing cores is clocked a little higher than the other when that other processor is not under much load – in this case, the boost to one of the cores is to 2394 MHz (9 x 266 MHz FSB frequency)[2].

Note: CPU clock frequencies are derived from the Front Side Bus frequency. Traditionally, the CPU frequency has been an integer multiple of the FSB frequency, though some Core 2 Duo processors support half-step multipliers such as 8.5 in this case.

Note: Performance State P3 is achieved not by slowing just the CPU clock, but rather by slowing the Front Side Bus (which necessarily slows the CPU clock by the same factor). This provides for even greater power savings (and even lower performance) than by just slowing the CPU clock alone. Intel calls this technique Dynamic FSB Frequency Switching[3].

Note: These aren't the only P-states that are technically possible with this processor. They just constitute the states being used by the Dell Latitude E6500 under test.

Note: P1 is sometimes referred to as HFM (Highest Frequency Mode) by Intel documentation. Likewise, P2 is sometimes referred to as LFM (Lowest Frequency Mode) and P3 is referred to as SLFM (Super Low Frequency Mode) with the "Super" referring to Dynamic FSB Frequency Switching.

Sometimes, transitions to lower-power P-states are made in order to decrease power consumption when processing demand is low. As soon as processing demand increases enough to warrant a higher-performance P-state, the necessary upclocking transition is made nearly instantaneously. Intel calls this power-saving technique Speedstep Technology[4]. But lower-performance P-states can also be used on a sustained basis for the purpose of lowering operating temperatures even under high processing demand.

B. Software-controlled Clock Modulation

This is a mechanism to reduce the effective clock rate of the CPU by slicing up time into tiny equal intervals and then only running the CPU clock (and hence, the processor) during a fraction of each time slice. The fraction can be one of eight settings 1/8 through 8/8 = 1. That is, the clock runs either 8/8 of the time (all the time), 7/8 of the time, etc. down to 1/8 of the time, depending on what setting is chosen by software. So in the extreme case, if the setting is 1/8, the processor is effectively only running at 1/8th or 12.5% of the speed it was before Software-controlled Clock Modulation was engaged[5].

Normal operating temperatures

In general, CPUs and GPUs (Graphics Processing Units) generate the highest temperatures in a PC. However, these chips are both designed to run without problem even at fairly high temperatures. Normal operating temperatures range from around 35°C when idle to the 80s or even 90s Celsius under heavy load. All processors of any type are designed to operate normally under a specified range of temperatures. For example, the Intel Core 2 Duo P8400 processor in the system under test is rated for normal operation up to 105° Celsius[6]. Though specifications for the NVIDIA Quadro NVS 160M processor are not available online, GPU chip technology is similar to CPU technology and GPUs are designed to operate in similar temperature ranges (sometimes at even higher temperatures than a system's CPU).

While high temperatures beyond the rated operating range can result in processing errors and extreme temperatures can result in physical damage, modern-day CPUs and GPUs are commonly designed with built-in temperature sensors, automatic clock throttling capabilities and emergency shutdown mechanisms. For example, the Intel Core 2 Duo P8400 processor under test is capable of automatically engaging both clock-throttling mechanisms described above in the event of overheating (once the internally measured temperature exceeds the rated maximum of 105° Celsius). The processor also has an emergency shutdown feature, which halts all operation (and requires power to be cut) once the internally measured temperature reaches around 125° Celsius[7].

How sustained clock throttling occurs at normal operating temperatures

In the system under test, once internal temperatures reach a given point (still well within normal operating range), the system is throttled first by transitioning to lower-performance P-states. As long as the temperature remains elevated enough, lower and lower performance P-states are activated in succession at 30 second intervals. Assuming the system starts in state P0, it first transitions to P1, then 30 seconds later to P2 and 30 seconds after that to P3. It's possible for the system to throttle down only to P1 or P2 and then hold there, foregoing any further throttling so long as the internal temperature declines enough. However, if the lowest-performing P-state is reached and the system still determines the temperature is too high, then it will begin Software-controlled Clock Throttling, successively cutting the effective CPU clock (and thus, the CPU's performance) in the available 1/8 increments, as previously described. This also happens at intervals of 30 seconds and continues as long as the system determines the temperature is too high. Once the system progresses through all available Software-controlled Clock Throttling settings, the system effects no further throttling. In total, there are 10 incremental steps, progressively applied at 30 second intervals, that are used to throttle the processing power of the system:

Throttling action	Effective processing power (frequency)
1. Transition from state P0 to P1	2261 MHz
2. Transition from state P1 to P2	1596 MHz
3. Transition from state P2 to P3	798 MHz (with FSB frequency cut in half)
4. Clock Throttling at 7/8 capacity of state P3	700 MHz (with FSB frequency still reduced by half)
5. Clock Throttling at 6/8 capacity of state P3	600 MHz (with FSB frequency still reduced by half)
6. Clock Throttling at 5/8 capacity of state P3	500 MHz (with FSB frequency still reduced by half)
7. Clock Throttling at 4/8 capacity of state P3	400 MHz (with FSB frequency still reduced by half)
8. Clock Throttling at 3/8 capacity of state P3	300 MHz (with FSB frequency still reduced by half)
9. Clock Throttling at 2/8 capacity of state P3	200 MHz (with FSB frequency still reduced by half)
10. Clock Throttling at 1/8 capacity of state P3	100 MHz (with FSB frequency still reduced by half)

While this aggressive degradation of the system's capabilities might be warranted if internal temperatures were so high that they might cause damage or malfunction, unfortunately in the system under test, these steps are activated at normal internal operating temperatures, in some cases even well below 70° Celsius. It's particularly

problematic that the user receives no notification regarding these drastic changes which severely impact performance. As a further complication, even though the throttling action brings down temperatures dramatically, performance levels are sometimes not restored for a long time, requiring a system reboot to clear the throttled condition.

The possible role of ACPI

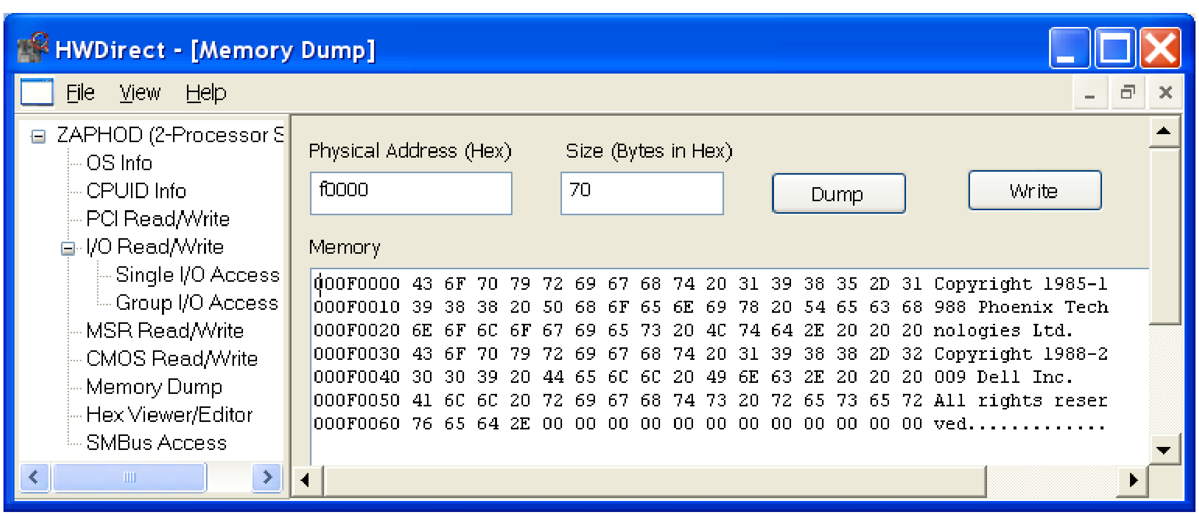
The Advanced Configuration and Power Interface is an interface specification standard promoted by Intel, Microsoft, Phoenix Technologies and other corporations which allows operating systems to communicate in a standardized way with physical hardware devices in a computer system. It provides a robust and uniform way for operating systems such as Windows to detect, monitor and configure a wide spectrum of devices including very low-level hardware such as power supplies, batteries, cooling fans and processor chips. In a system that is fully compliant with ACPI, configuration of such devices is performed exclusively through the operating system via the ACPI interface[8].

The throttling behavior observed would be consistent with an ACPI "passive cooling" policy. Passive cooling is defined in ACPI as a method whereby the "OS reduces the power consumption of devices at the cost of system performance to reduce the temperature of the machine." In the standard, a temperature threshold _PSV is defined beyond which passive cooling is engaged. ACPI suggests providing hysteresis by changing _PSV to a lower temperature once the initial threshold is reached and using that new value as a target temperature while passive cooling is engaged. An equation is even suggested for how to adjust the system's performance to achieve the lower temperature (using vendor-provided constants _TC1 and _TC2). The standard provides for periodic polling of the system temperature for the purposes of passive cooling using a sampling period _TSP[9].

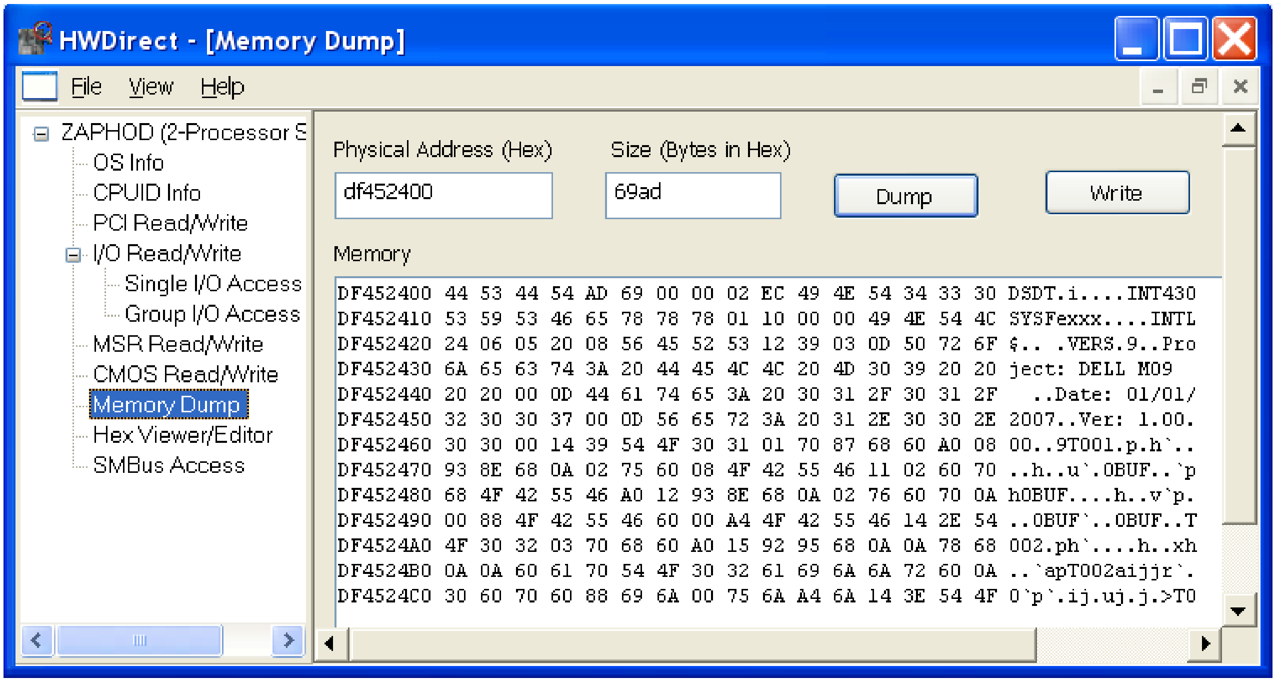
While it's not practically possible to independently confirm conclusively that this system is fully compliant with ACPI, or even that this system's performance loss is being effected through ACPI (particularly since passive cooling is not an absolute requirement for ACPI compliance), there are a number of facts that strongly suggest both may be the case.

1. The strings "_TC1", "_TC2", "_PSV" and "_TSP", which are particular to the "passive cooling" option of the ACPI standard, all appear in the file ACPI.SYS file under C:\WINDOWS\SYSTEM32\DRIVERS.

2. Microsoft (the operating system vendor), Intel (the CPU vendor) and Phoenix Technologies (the BIOS vendor) are all promoters of the ACPI standard. Phoenix Technologies can be confirmed as the BIOS vendor for the Dell Latitude E6500 using a BIOS scan at address 0xF0000 (see screen capture image below)



3. The Phoenix BIOS in the Dell Latitude E6500 contains the expected ACPI tables and code required by the standard (Root System Description Pointer at address 0xFB9C0, Extended System Description Table at address 0xDF451E00, Fixed ACPI Description Table at address 0xDF451C9C, Differentiated System Descriptor Table at address 0xDF452400, Secondary System Description Table at address 0xDF45032D, etc.). A screen capture image of the beginning of the Differentiated System Descriptor Table appears below:



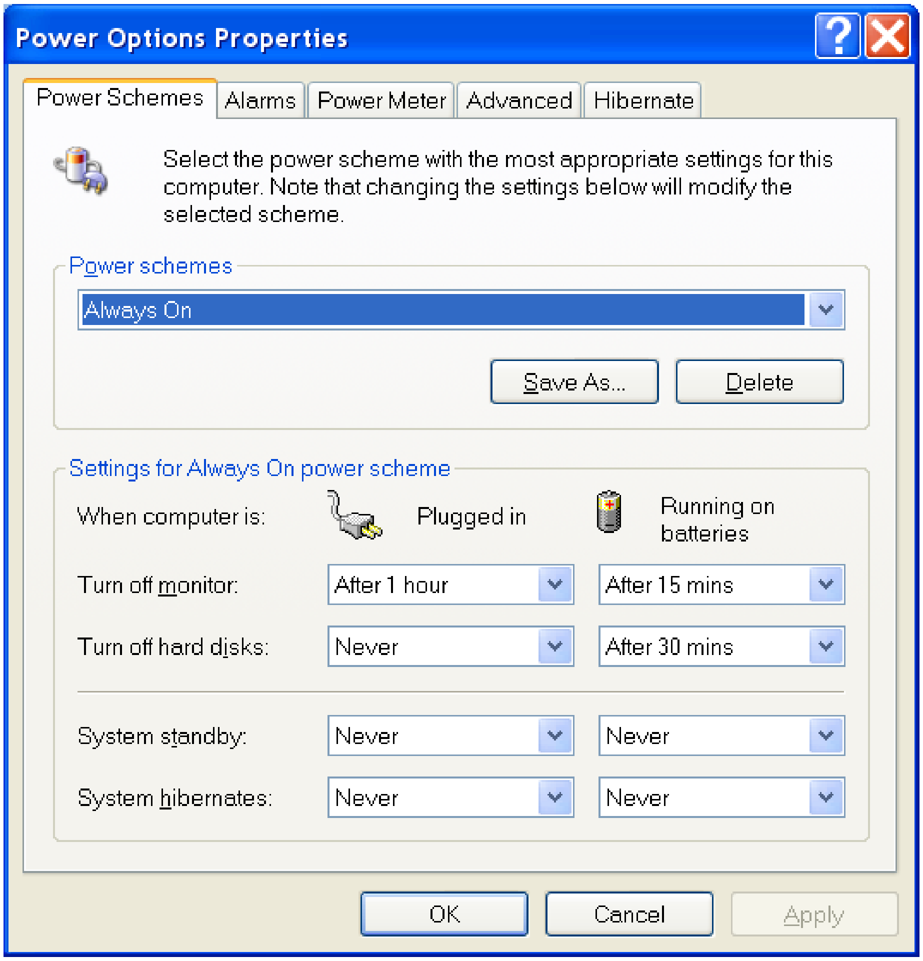
<u>Illustration: maximum performance loss due to clock throttling at normal operating temperatures</u>

Following is an illustrated example of exactly how the Dell Latitude E6500 is progressively throttled to the point that it is operating at less than 5% of processor capacity. Screen captures are used to chronicle the process.

First, here is a more complete description of the test system:

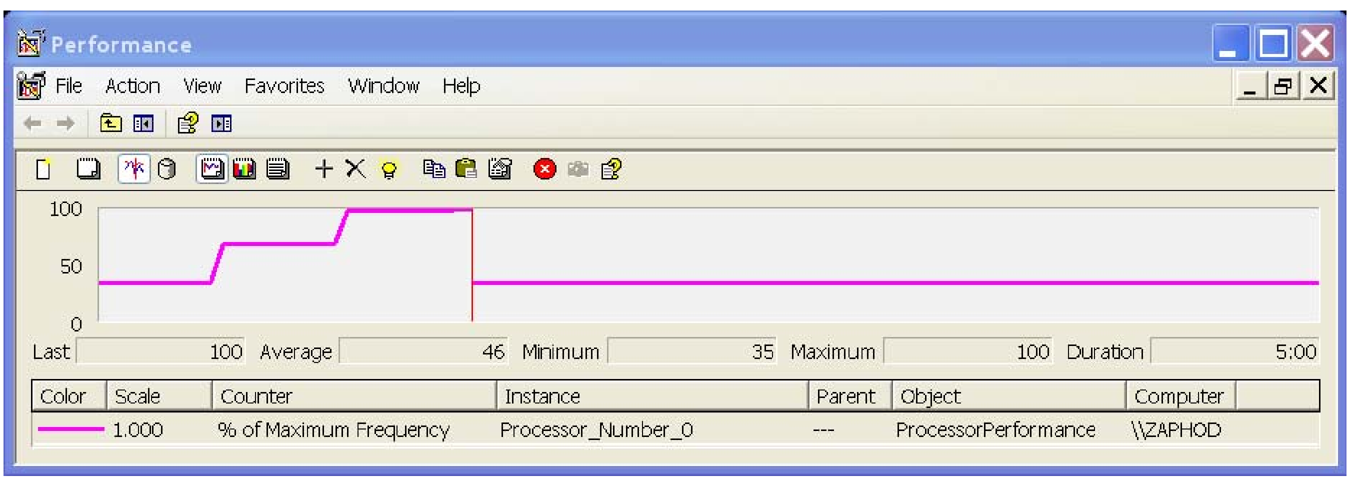
<u>Dell Latitude E6500</u>
Intel Core 2 Duo P8400 processor, 2.26Ghz
4GB RAM
NVIDIA Quadro NVS 160M graphics
Dell BIOS version A13
E/Port Plus Advanced Port Replicator (Docking Station)
dual Dell E248WFP monitors (only one monitor used in some tests)
Windows XP Service Pack 3

Configuration note: A key adjustment for these tests was to switch to the “Always On” power scheme in the “Power Options” Control Panel window. Choosing the “Always On” power scheme disables Intel’s Speedstep feature which temporarily downclocks the processor (by transitioning among the various P-states) when it is idle. To avoid confusion about what caused a P-state change, it was important to disable Speedstep.



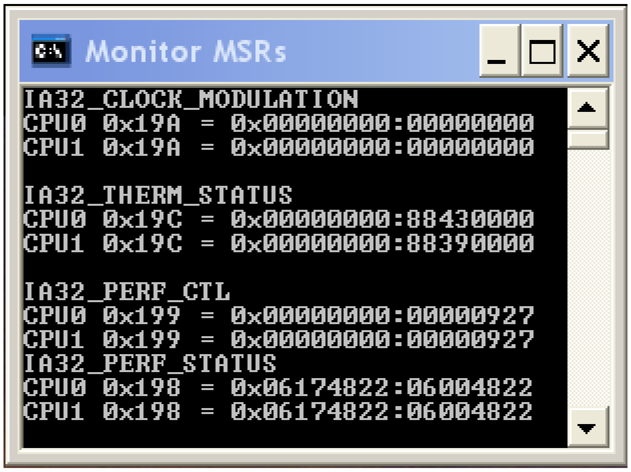
Maintenance note: The system was checked for internal dust that might clog cooling systems prior to testing.

Following is a description of the utilities used in testing.



Above is Windows XP Performance Monitor (perfmon.exe), configured to monitor and graph the operating frequency of one of the two processor cores present in the Intel Core 2 Duo processor. This value is shown as a percentage of maximum frequency. This graph will indicate whenever the performance state (P-State) changes. The possible values are 100%, 99% (corresponding to state P1), 70% (corresponding to state P2, at about 1600 MHz) and 35% (corresponding to state P3, which is about 800MHz). In this screen capture, perfmon is showing a rise over time from 35% capacity to 100%, with the performance stepping up every 30 seconds. Though the graph only shows the trace for one of the processor cores, both generally operate at exactly the same frequency

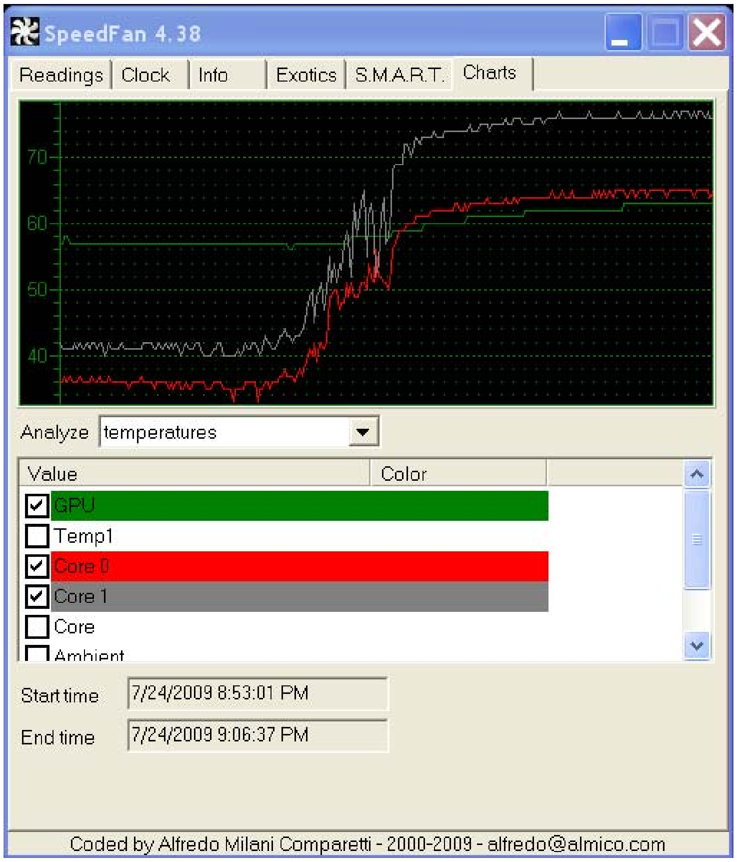
in practice. It is important to note that perfmon updates its graph differently than the other graphing utilities used in these tests. Perfmon uses a thin vertical bar cursor that moves from left to right providing the latest reading at the cursor itself as it moves across the screen (the graph itself remains stationary over time). The other utilities (SpeedFan, Task Manager and DPC Latency Checker, described below) produce a graph that moves in its entirety from right to left with the most recent value shown on the right edge.



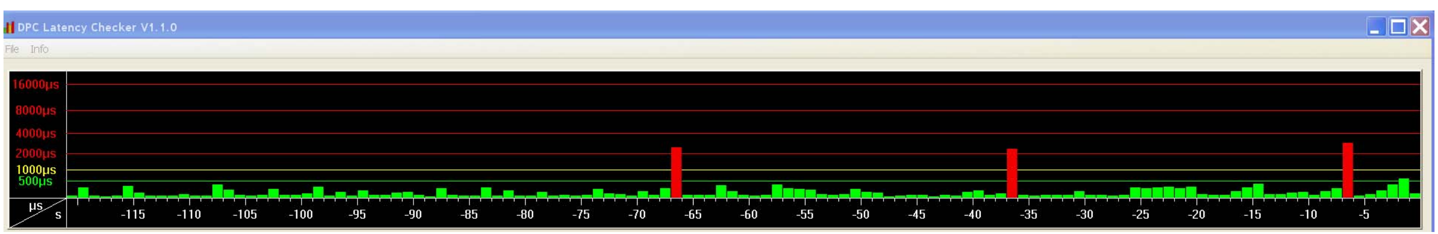
The window titled "Monitor MSRs" is a looping CMD.EXE batch file that displays key Model-Specific Registers (MSRs) once every second. MSRs are memory storage locations within the CPU itself that allow inspection of key operating parameters. The batch file uses msr.exe from the freely available "Performance Inspector" package. The values displayed (once decoded) indicate the current configuration for Software-controlled Clock Modulation (IA32_CLOCK_MODULATION), processor core temperature (IA32_THERM_STATUS), last performance state configuration requested (IA32_PERF_CTL) and the current performance state in effect (IA32_PERF_STATUS). These values can be decoded using Intel documentation available online[10]. The batch file is revealed in Appendix A.

The Performance Inspector package is available online at: http://perfinsp.sourceforge.net

---



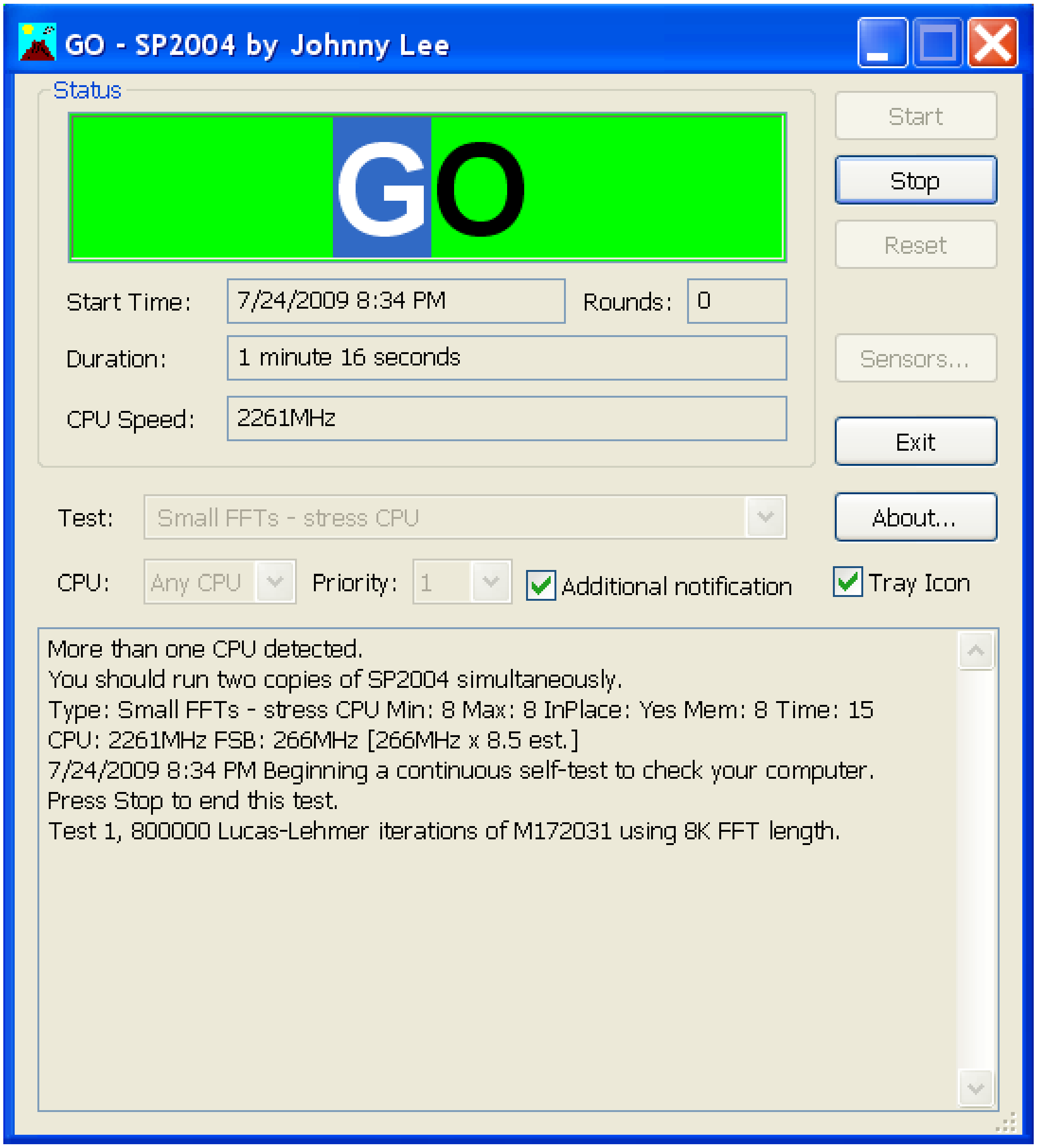
Speedfan is a freeware utility that's used to monitor and graph temperatures for both CPU cores and the NVIDIA GPU, as shown here. It's available online at http://www.almico.com/speedfan.php .



DPC Latency Checker monitors and graphs the delay that's present in the Windows DPC (Deferred Procedure Call) queue. Deferred Procedure Calls are generally software routines that process output or input data (say, to or from a network or a DVD drive). If you know what an ISR (Interrupt Service Routine) is, a DPC is like a lower-priority type of ISR. DPCs are often used during playback of audio and/or video and if the latency (a fancy way of saying "delay") in the queue of DPC requests gets too long (that is, the queue gets backed up enough), you can wind up with audio that crackles or breaks up and video that hitches, no matter how fast your system is. If the delay gets very long, audio or video can become unintelligible or stop and other important processes can become crippled as well. It also turns out that this utility provides an excellent way to determine exactly when clock throttling events occur. As long as a system normally has a low DPC latency, this utility easily exposes and records each clock throttling event (both downclocking and upclocking). Each event appears as an easily identifiable spike in DPC latency that generally stands out among the surrounding samples (as in the screen capture above). The fact that clock throttling events happen only at intervals of 30 seconds aids in identification of these events.

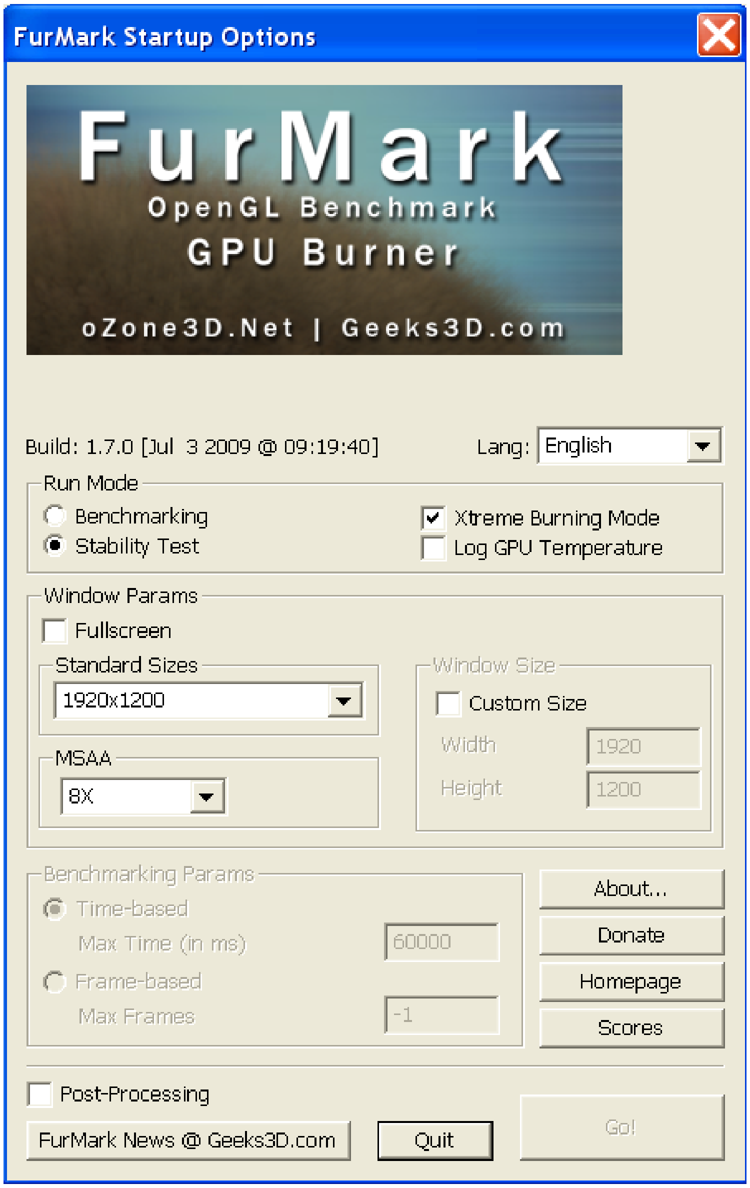
DPC Latency Checker is available online at: http://www.thesycon.de/deu/latency_check.shtml

---

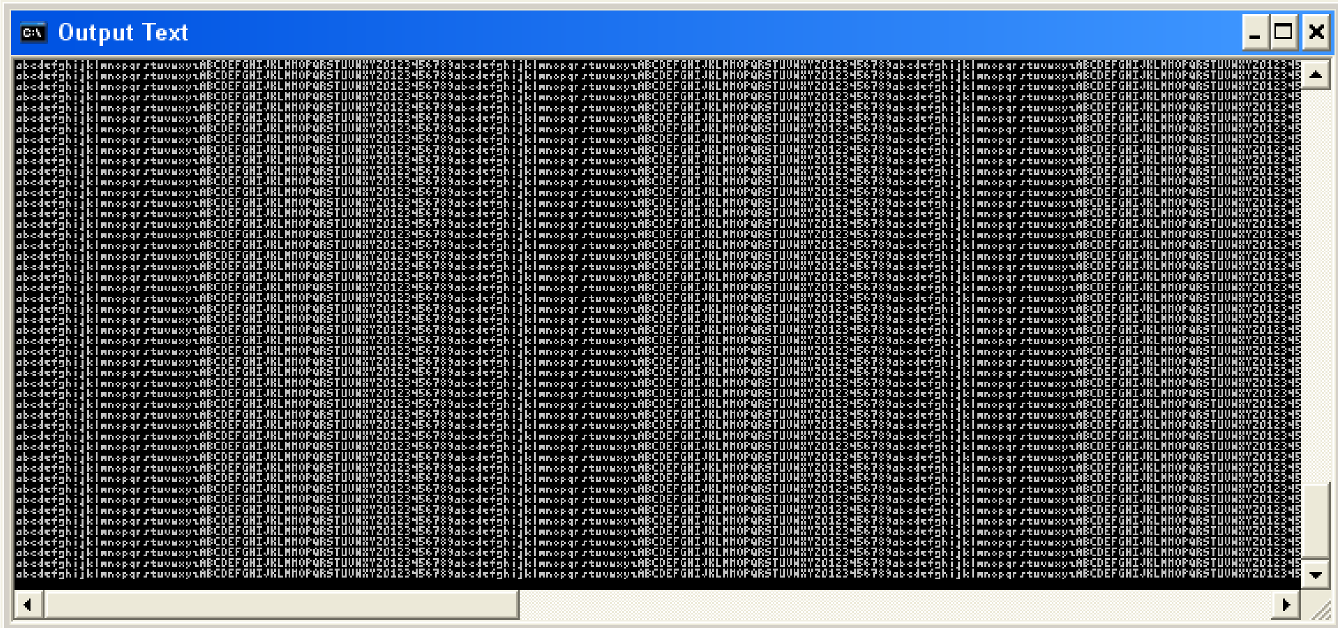


Stress Prime 2004 is a freely-available CPU-stressing utility. Each instance of this utility stresses a single processor core, so two instances are needed to fully exercise the Intel Core 2 Duo processor package. This utility is available online at http://sp2004.fre3.com/index.htm

---



FurMark is a freeware graphics processor benchmarking and stressing utility. It is used in these tests to exercise the NVIDIA graphics processor in order adjust the GPU temperature in a controlled manner. It turns out that the CPU clock throttling in the Dell Latitude E6500 is actually correlated more closely to GPU temperature than CPU temperature (at least on this system with discrete NVIDIA graphics – Dell's Latitude E6500 can also be purchased from Dell with an integrated Intel graphics option, which was not tested for this article). FurMark is available online at: http://www.ozone3d.net/benchmarks/fur



The “Output Text” window is a very simple 4-line looping CMD.EXE batch file that just creates scrolling lines of text ouput. When small raster fonts are used and large windows of this process are placed on the Desktop, this can be used to exercise the GPU and therefore adjust its temperature in a controlled way. The batch file is revealed in Appendix B.

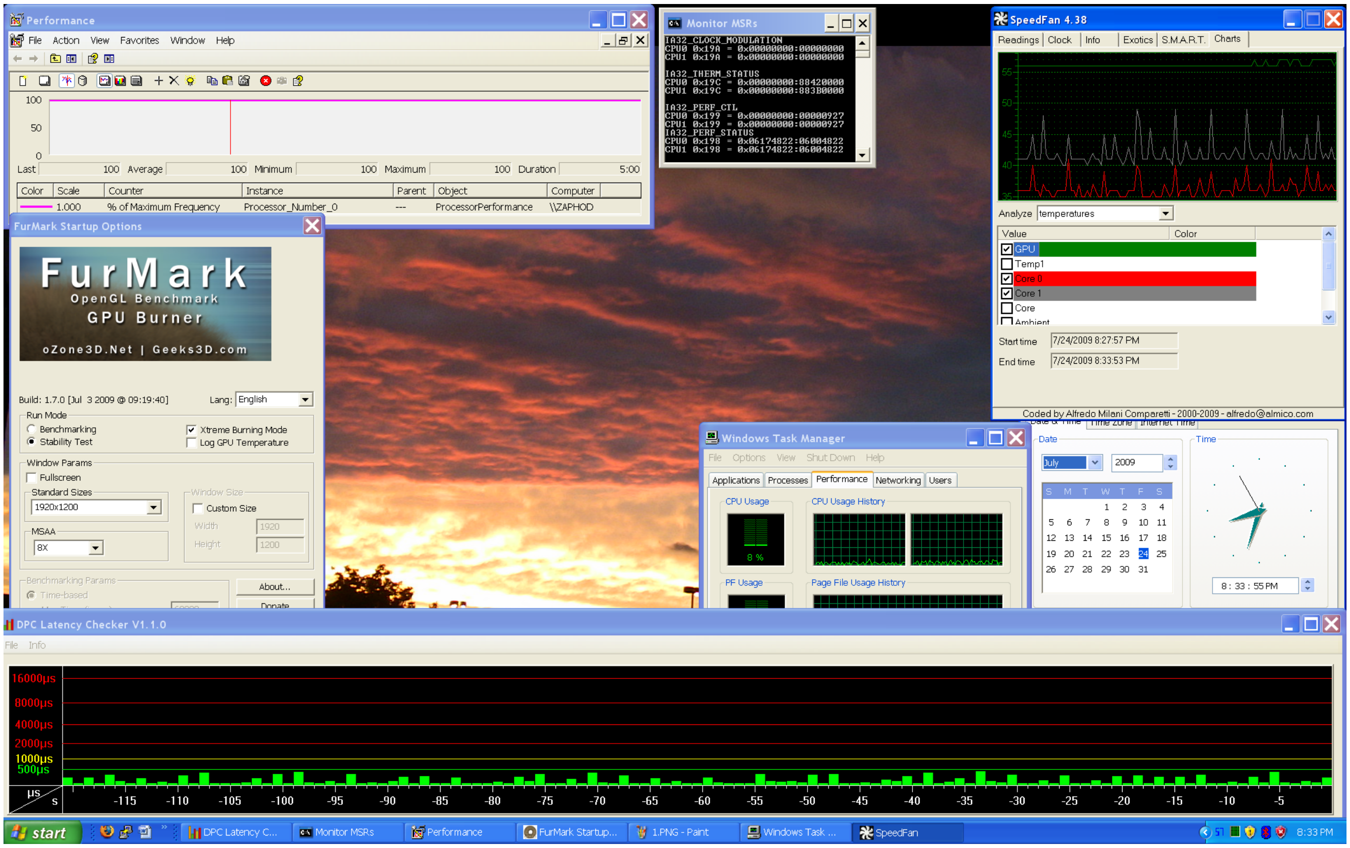
Finally, the Windows XP standard utilities Clock and Task Manager are used to document date/time and CPU utilization. Note that for the testing screen shots show below, you can calculate time differences between them by comparing the Clock images. The Windows XP standard utility Calculator is used as a way to exercise the CPU in some tests.

For reference, in all the tests for this article, Perfmon samples occured every 3 seconds, Speedfan samples every 3 seconds, Task Manager every 2 seconds and DPC Latency Checker every 1 second.

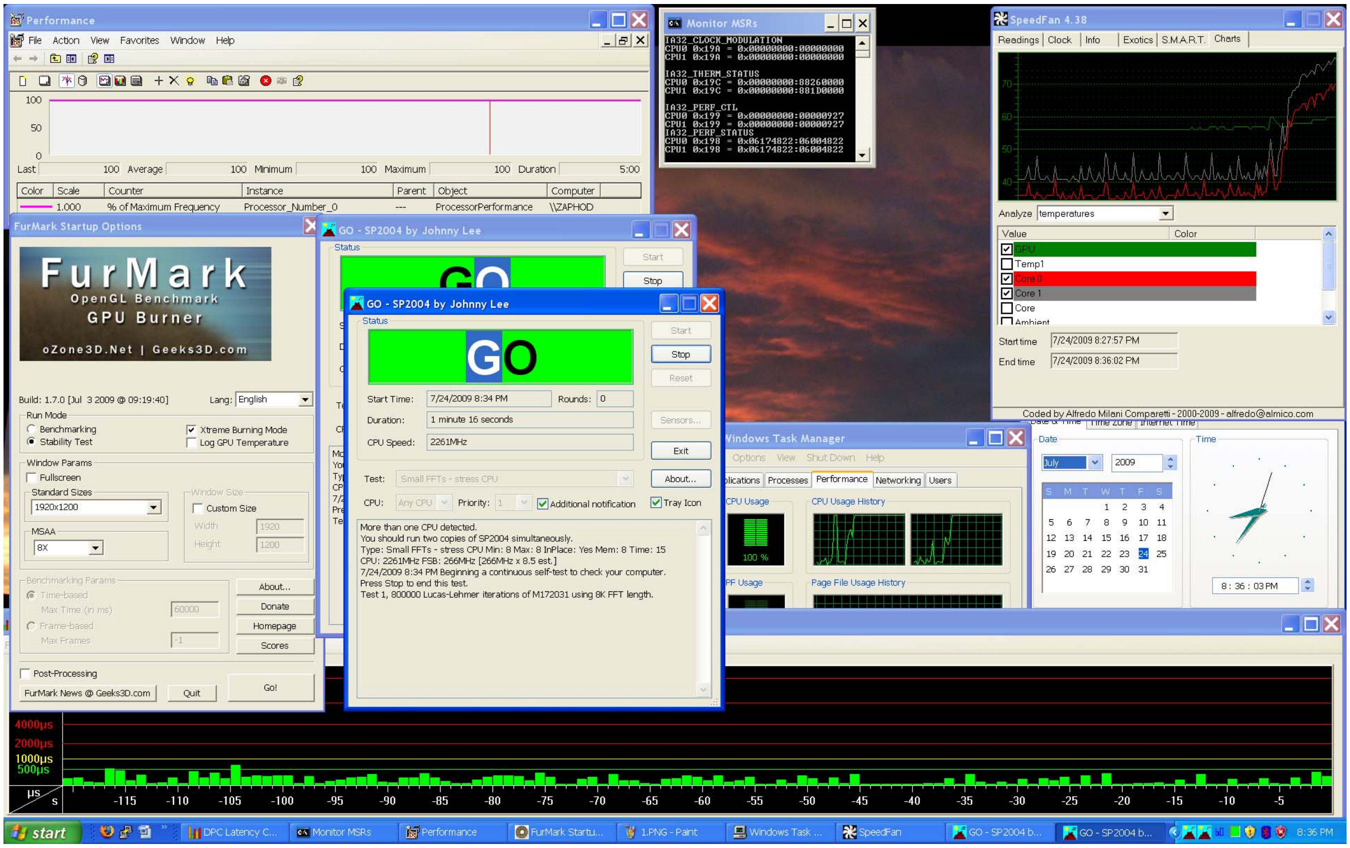
In the first test below, which spans a period of one hour and is documented with 31 screen captures, the GPU is exercised until its temperature reaches just under 70°C, the temperature is maintained as the system’s performance is progressively throttled through 10 different steps, one every 30 seconds, until maximum throttling is reached. Then the system is allowed to cool and the performance reductions are seen to be reversed, also every 30 seconds. Then the system’s CPU is exercised, rather than the GPU, until the temperatures of the two processor cores reaches an average just above 70°C where, again, the system is observed to throttle performance progressively every 30 seconds. However, this performance degradation is not reversed even after allowing the system to cool for 20 minutes. A short period of cooling with an external desk fan is used to return the system to full processing power.

For the following test, the Dell Latitude E6500 was docked to the E/Port Plus Advanced Port Replicator. A single Dell E248WFP monitor (connected to the E/Port Plus) was used, operating at its native resolution of 1920x1200.

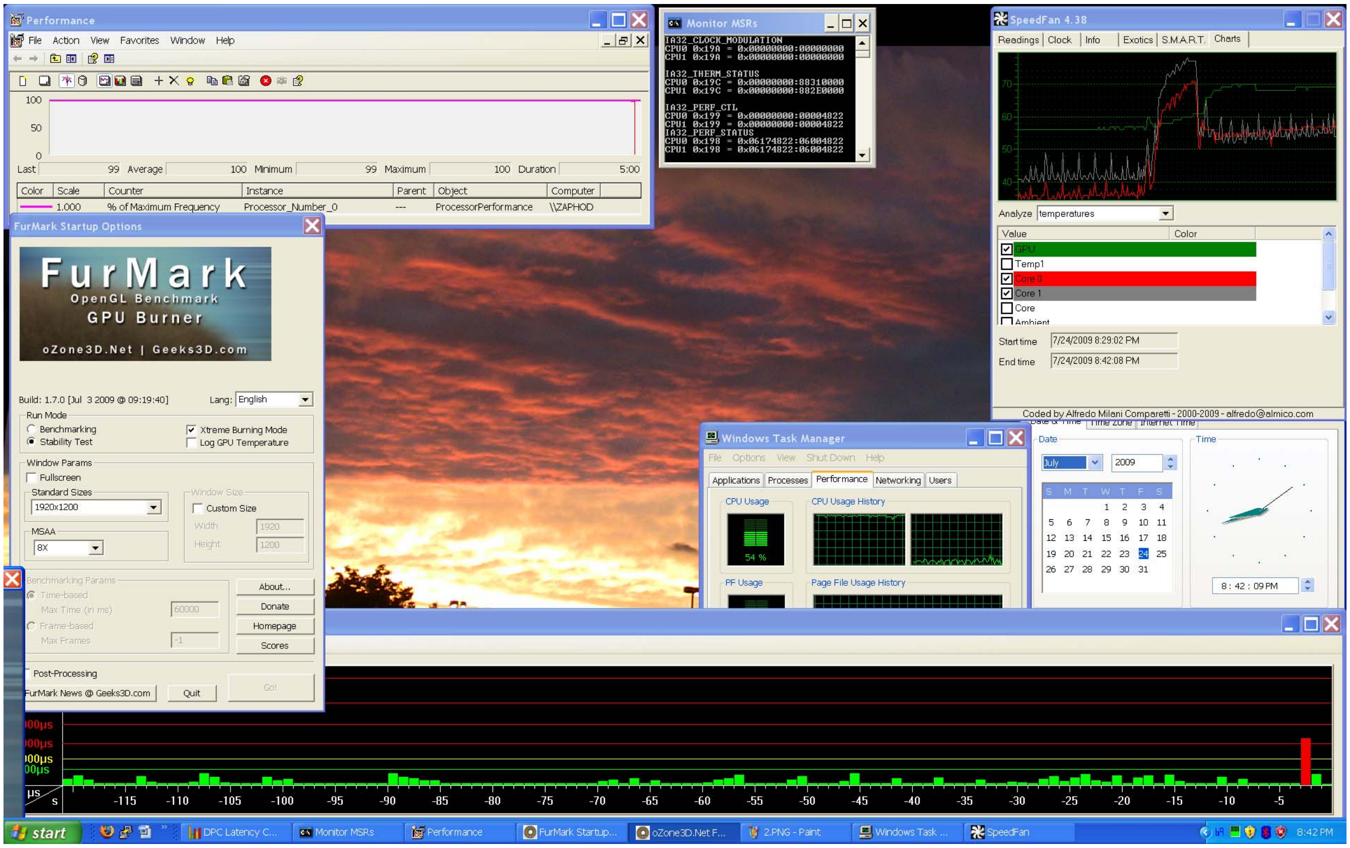
Note that all included screen capture images are *not* highly compressed. If you are viewing this document in its original form, this should allow for excellent legibility when one zooms in on any portion of the image.



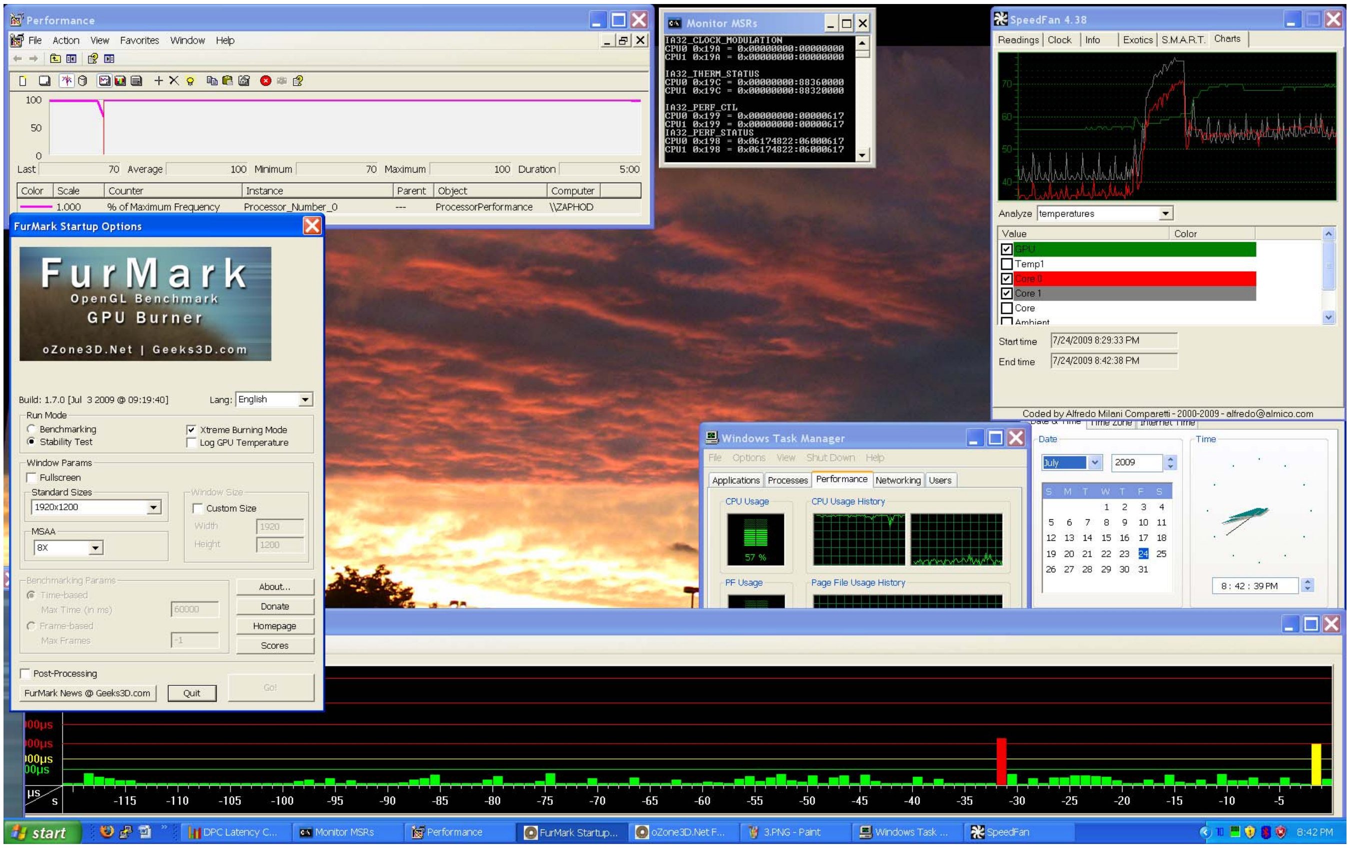
In this first screen capture, taken before the testing begins, it can be seen that the system is at full processor capacity (100% of maximum frequency according to perfmon in the upper left corner), the GPU temperature is about 56°C and CPU core temperatures are averaging around 36°C and 42°C (according to SpeedFan in the upper right), the system has been idle for some time (according to Task Manager), DPC latencies are under 500 microseconds (according to DPC Latency Checker along the bottom) and FurMark has not yet been started. Ambient room temperature at the time of this test is 28.5°C.



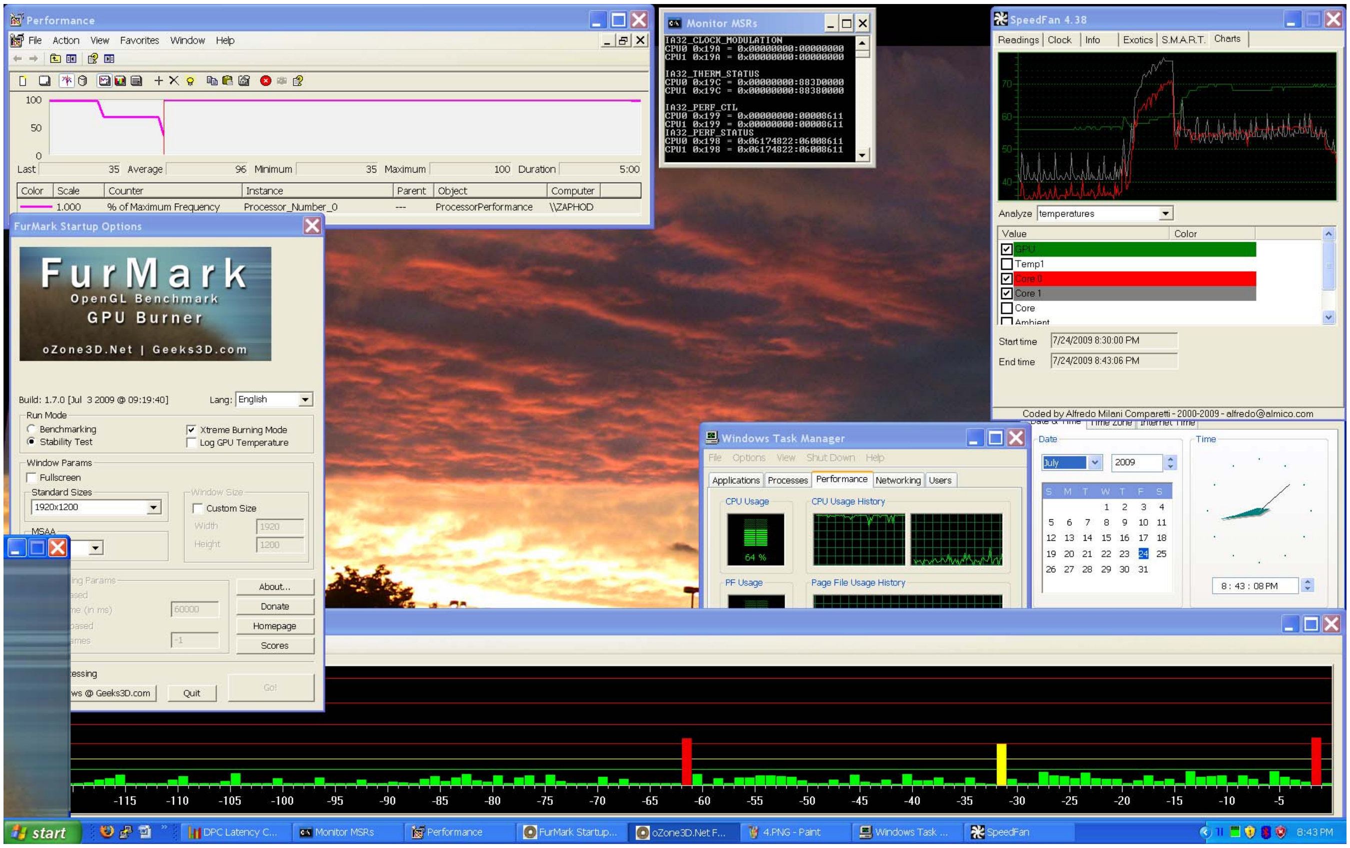
To begin with, demonstrating a point that will be expanded on later, two instances of Stress Prime 2004 are started to heat up the processor cores until they're both above 70°C (when this screen capture was taken). This was done to demonstrate that no throttling takes place when this happens. The next step will be to heat up the NVIDIA Quadro NVS 160M to just under 70°C, where we'll see that throttling ***does*** take place.



This capture was taken about 6 minutes after the Stress Prime 2004 processes were stopped and a FurMark process was started to stress the GPU (a corner of FurMark's graphics rendering window can be seen at the extreme lower left). CPU core temperatures have dropped to about 55°C and the GPU temperature is hovering around 69°C. The first throttling action has just taken place. Perfmon in the upper left shows a slight dip in the graph trace at the vertical bar cursor toward the far right of its window and the "Last" measurement is shown as 99% now instead of 100%. This indicates the transition to Performance state P1 from state P0. DPC Latency Checker, along the bottom, recorded the switch as having occurred 3 seconds prior to the capture. All throttling events, in fact, are marked by such conspicuous latency spikes. The Intel Core 2 Duo's IA32_PERF_CTL Model-Specific Register now shows a different value in the "Monitor MSRs" window, indicating that a performance state change was requested of the CPU.



Here the second throttling event takes place exactly 30 seconds after the first. All throttling events, in fact, are observed to take place only at 30 second intervals. This is consistent with an always-present running process that polls temperatures and takes throttling action every 30 seconds. Perfmon shows the decline in performance to 70% and IA32_PERF_CTL indicates the new P-State change request.



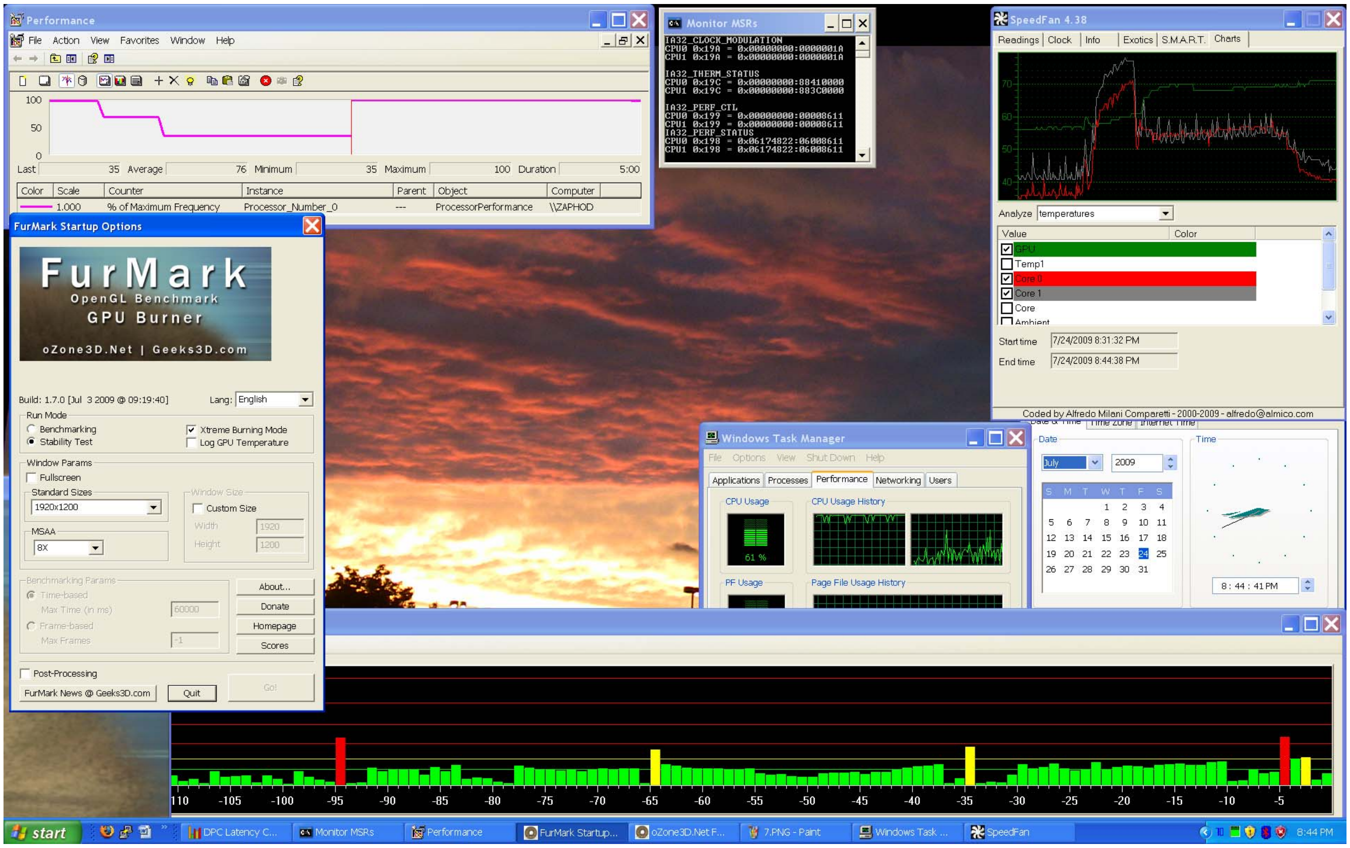
30 seconds later, the 3rd throttling event takes place. Perfmon shows %processor utilization is now down to 35% and IA32_PERF_CTL shows yet another value corresponding to the request for a change to performance state P3 (the lowest-performance P-state). Note that the position of the FurMark rendering window in the lower left is being shifted from time to time to make small adjustments to the GPU load so that the GPU's temperature is maintained around 69°C. The more of FurMark's rendering window is actually exposed on the Desktop, the higher the demand on the GPU.

At this point, the system is already at the lowest performing P-State (P3), so at this next 30 second interval, Software-controlled Clock Throttling begins. The CPU's IA32_CLOCK_MODULATION MSR is now showing that modulation is engaged and throttling is now at 87.5%. The register codings are published in Intel documentation available online. The values for IA32_CLOCK_MODULATION are as follows[11]:

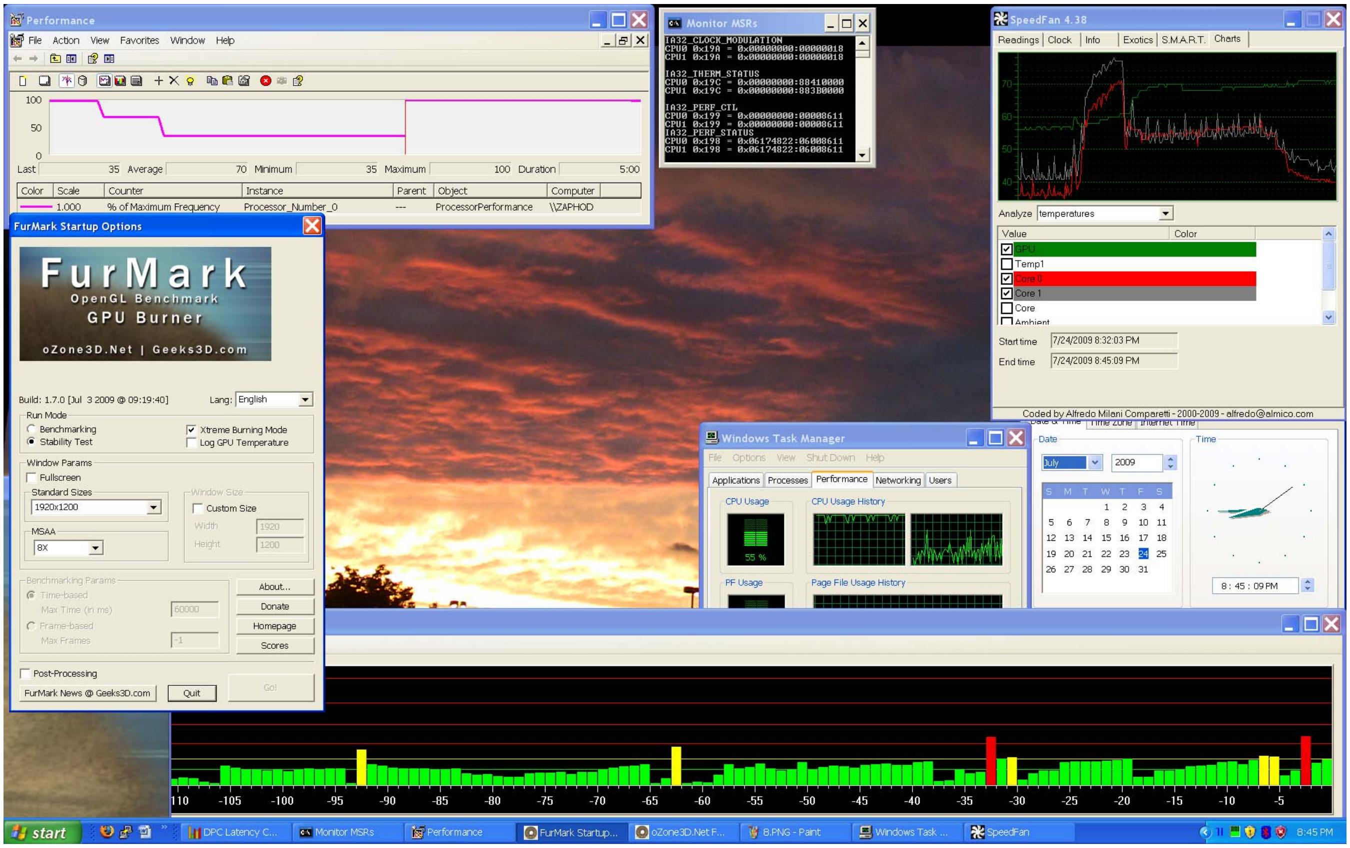
IA32_CLOCK_MODULATION values (in hexadecimal)	Software-controlled Clock Modulation level (percentage of time that the CPU clock is running)
0	Off (100%)
1E	87.5% (7/8)
1C	75% (6/8)
1A	62.5% (5/8)
18	50% (4/8)
16	37.5% (3/8)
14	25% (2/8)
12	12.5% (1/8)

CPU core temperatures are shown to be dropping because of the combined effect of downward P-state transitions and Software-controlled Clock Modulation.

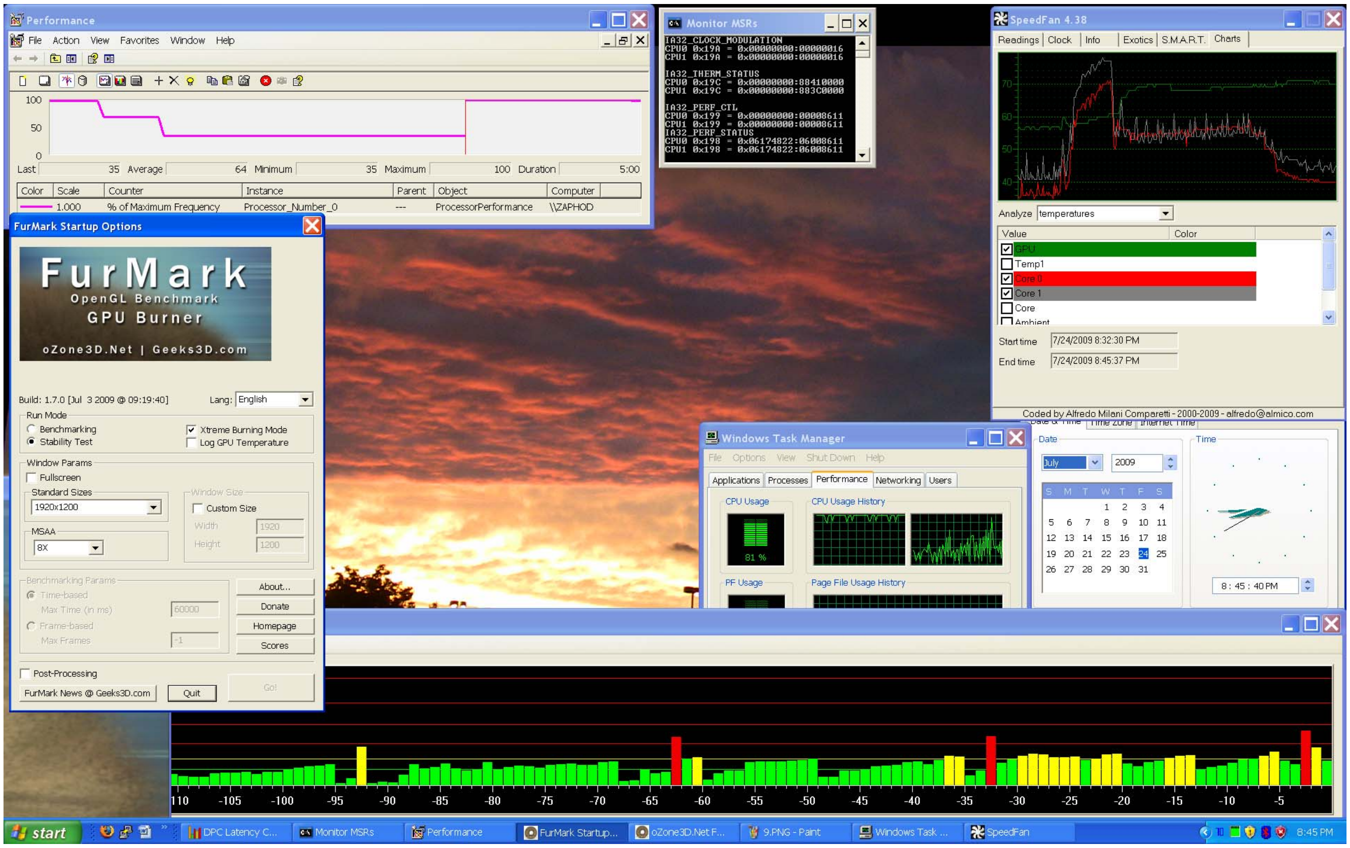
At the next 30 second interval, clock modulation increases to the next step so that processor performance is downgraded another 12.5% to 75%, as indicated by the new value for IA32_CLOCK_MODULATION. While CPU core temperatures are dropping due to the downclocking, the system's performance is also declining. The system becomes increasingly sluggish and unresponsive. Average DPC latency is also increasing significantly, as shown by the graph along the bottom, a concrete indicator decreasing system performance.



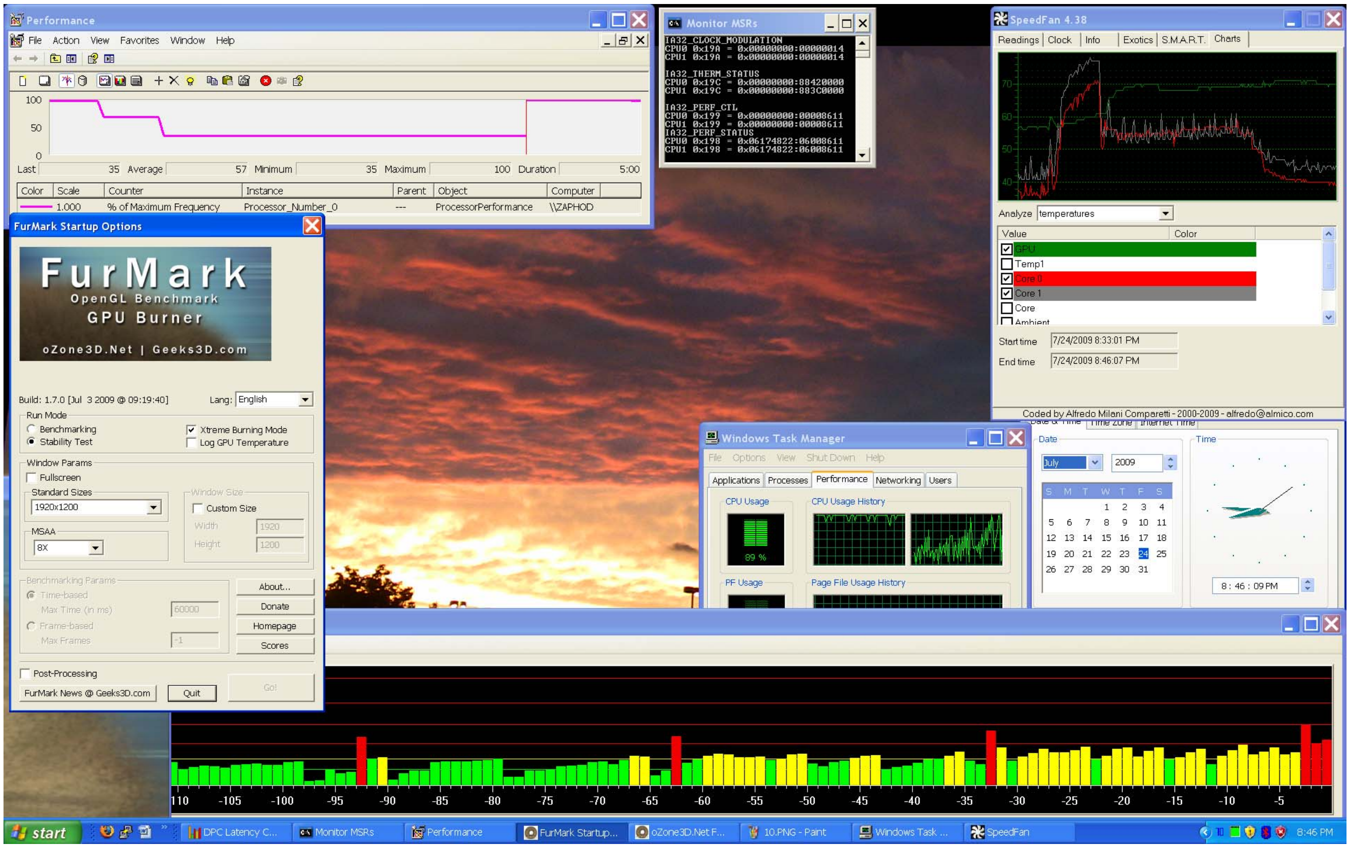
Right on schedule, clock modulation is escalated to the next level, now leaving the system at 62.5% of state P3's performance. Note that while CPU core temperatures have fallen dramatically, almost to idle levels before the testing began, the throttling continues unabated, demonstrating a strong correlation between throttling engagement and elevated GPU temperatures and little correlation with CPU temperature (though some correlation with CPU temperature is demonstrated later in this test session).



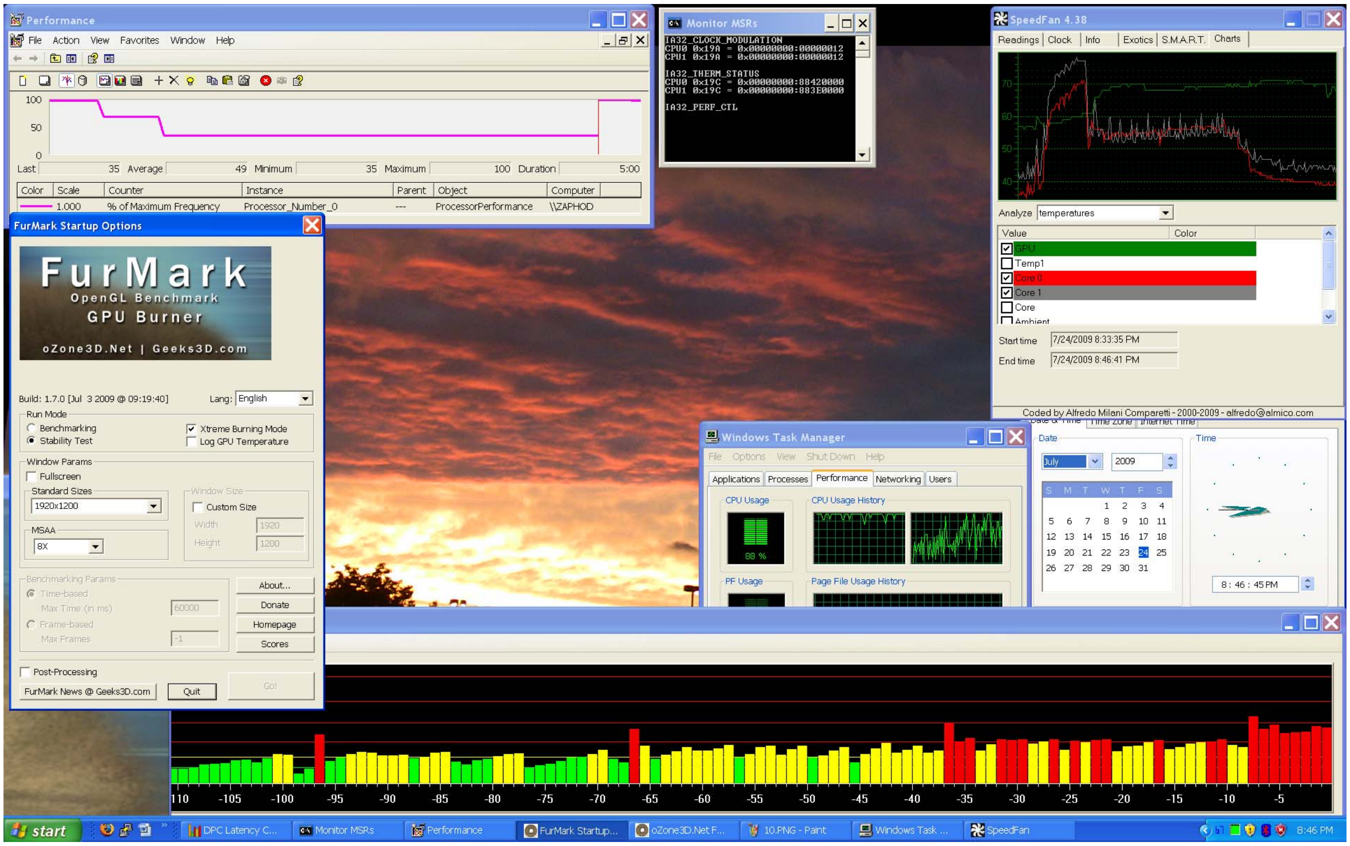
Throttling escalation continues. DPC latencies are approaching 1ms on average.



As the system performance continues to decline, baseline DPC latencies begin to take off as clock modulation is escalated (now down to 37.5% of state P3's performance).



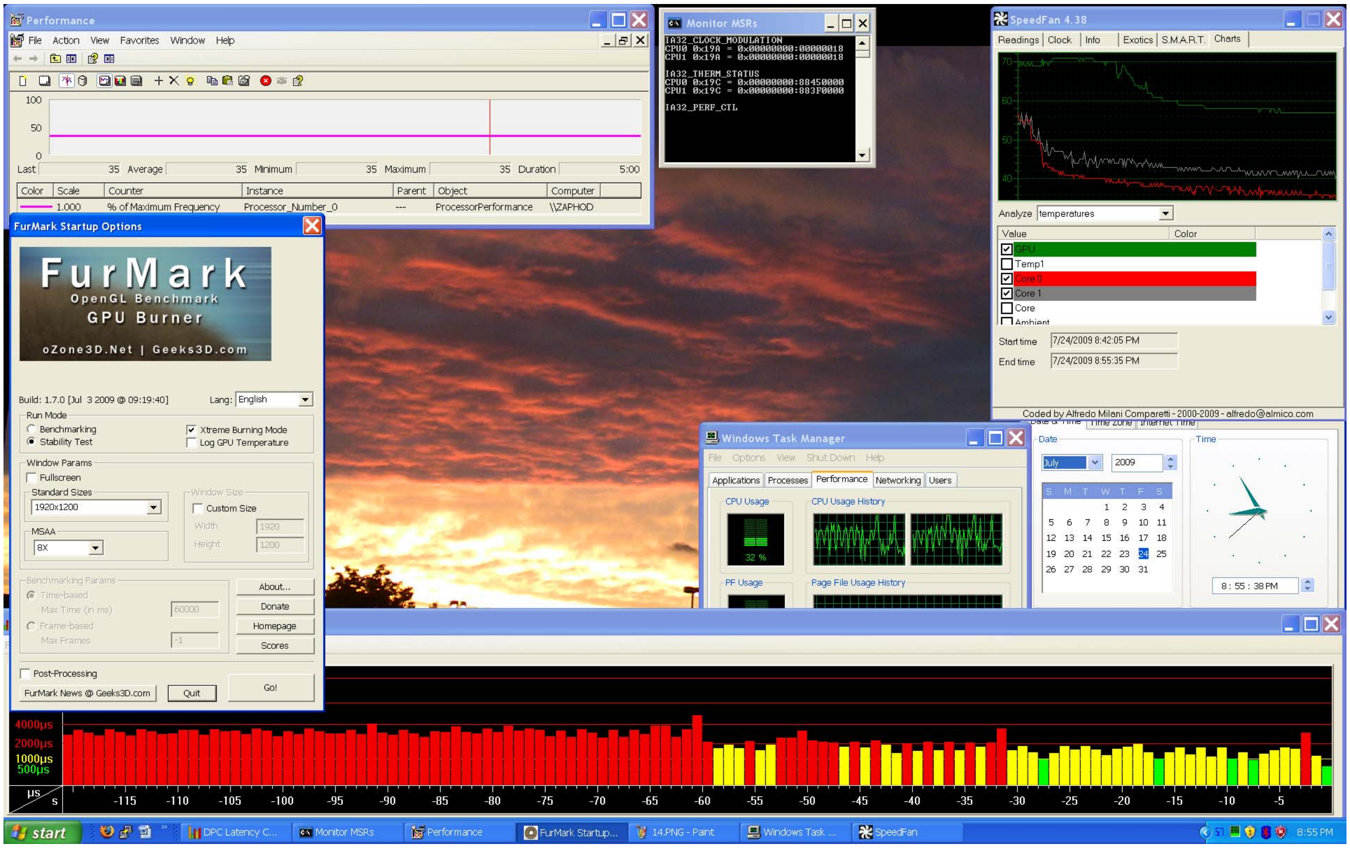
There is one more throttle event left.



The system is now throttled to 12.5% of state P3's performance (which is a state that was already downgraded to 798MHz with a half-speed Front Side Bus). The rough equivalent processing power for this 2261MHz system is now around 100MHz, according to numbers already highlighted from Intel's publications regarding these throttling features. However, throughout the entire throttling process, no internal temperature exceeded 70°C, itself a routine operating temperature. At this point, the FurMark rendering process was stopped to see how the system would recover.

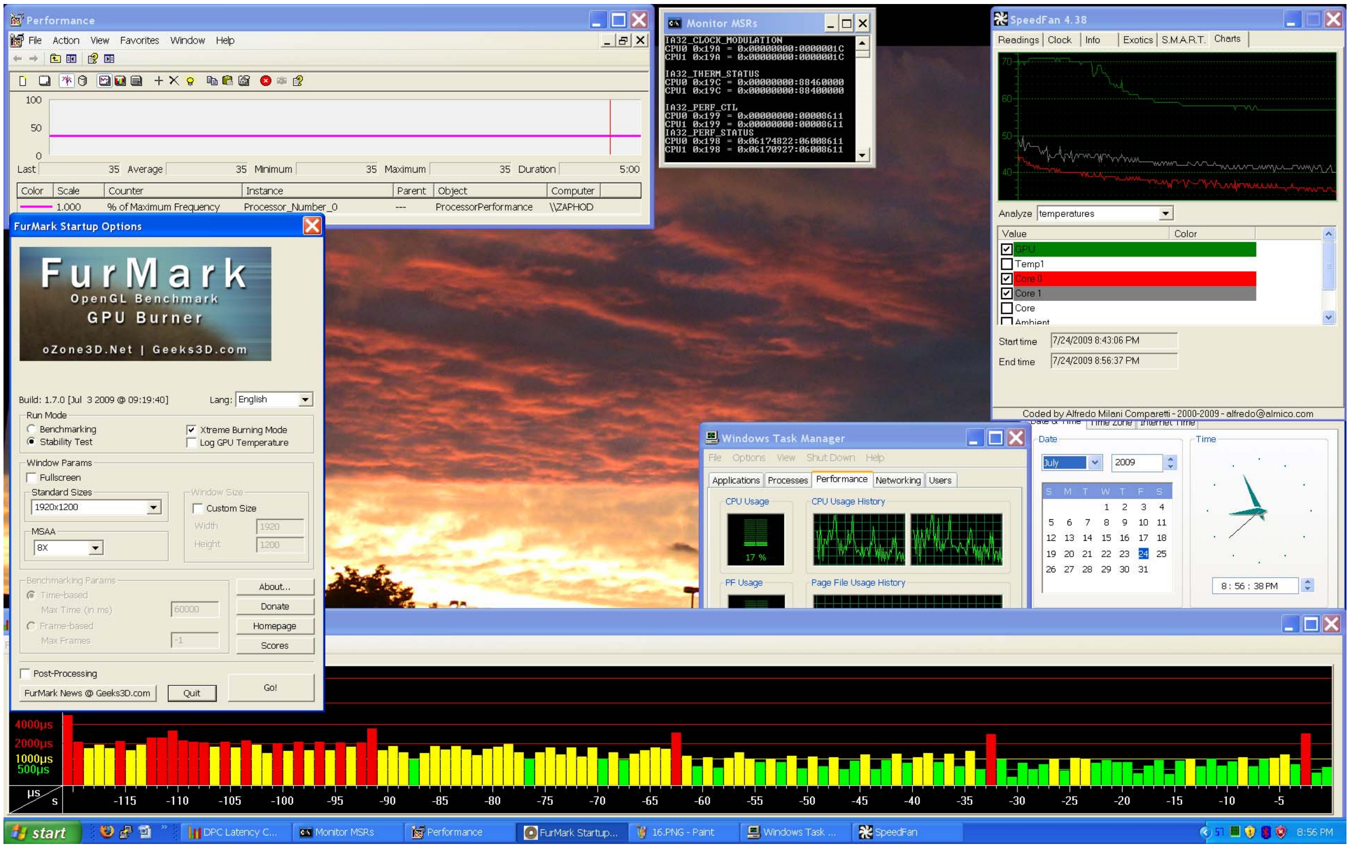
About 10 minutes later after the system slowly cooled off (chronicled in the SpeedFan graph in the upper right), reversal of the throttling began. The IA32_CLOCK_MODULATION register confirms the throttling transition to 25% (from 12.5%). The accompanying DPC latency spike (smaller than usual in appearance, but still substantial since the graph scale is logarithmic) records that the transition occurred 4 seconds before this screen capture.

The next few captures show the transitions, every 30 seconds, reversing the Software-controlled Clock Modulation. Now at 37.5%.

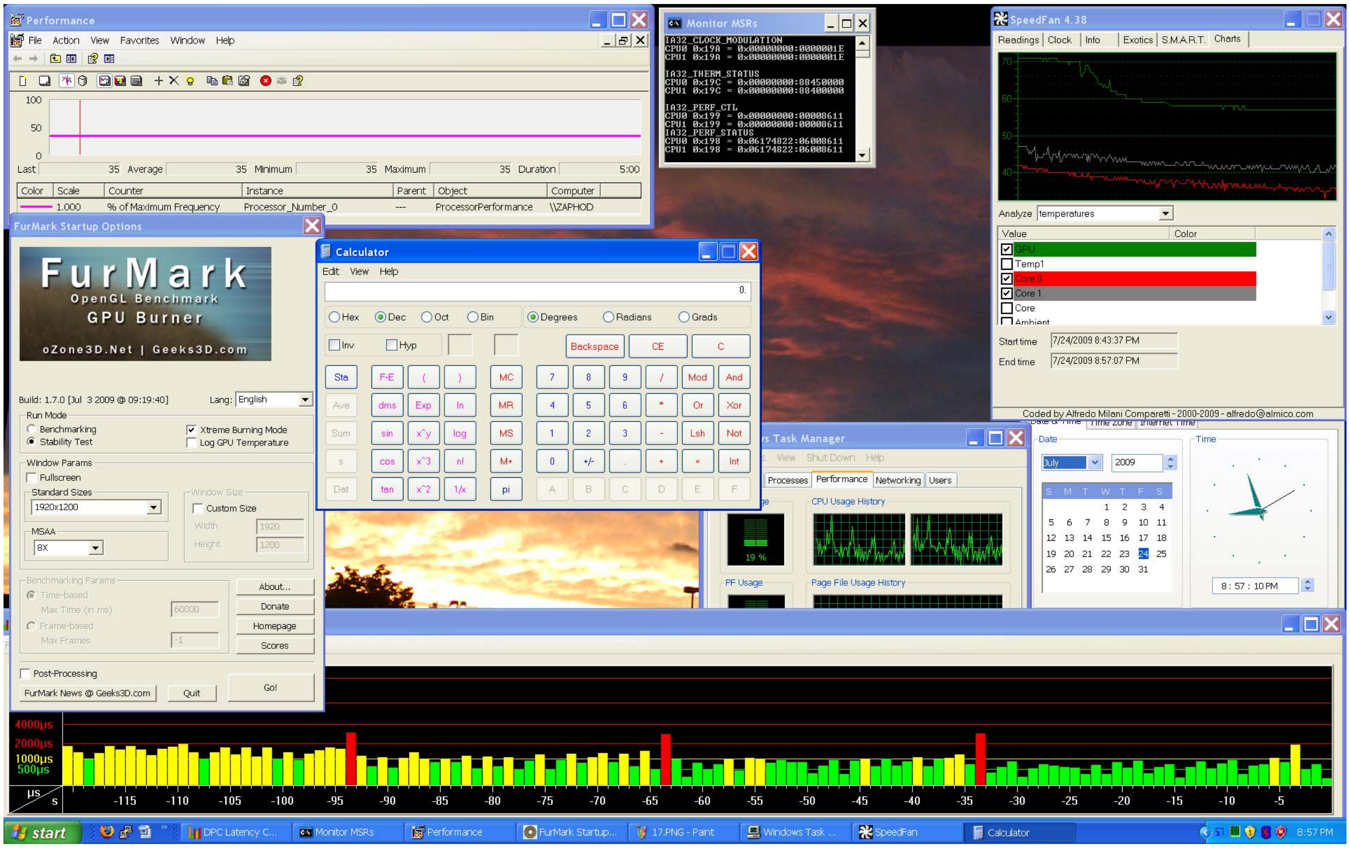


Now at 50% modulation.

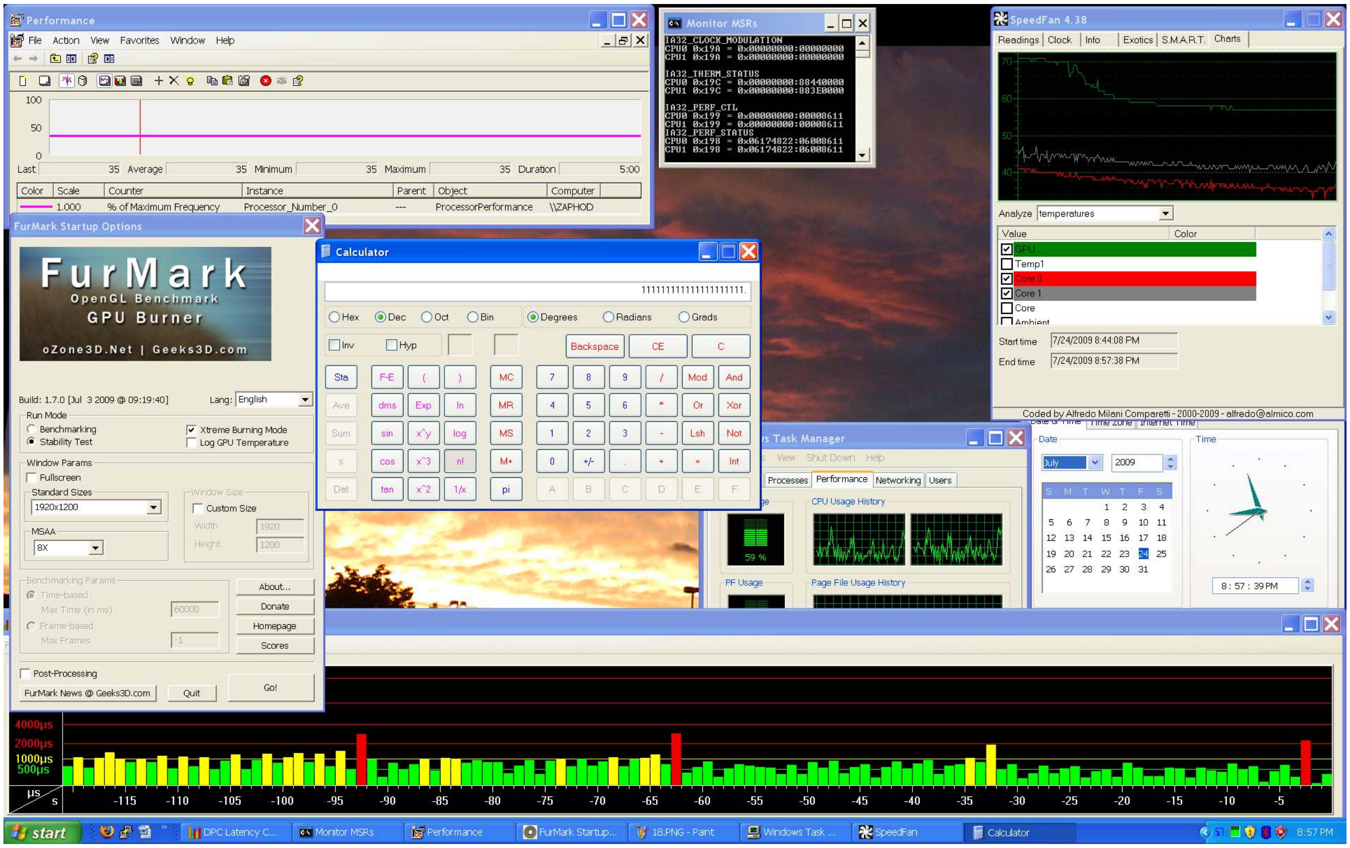
Now at 62.5% modulation.



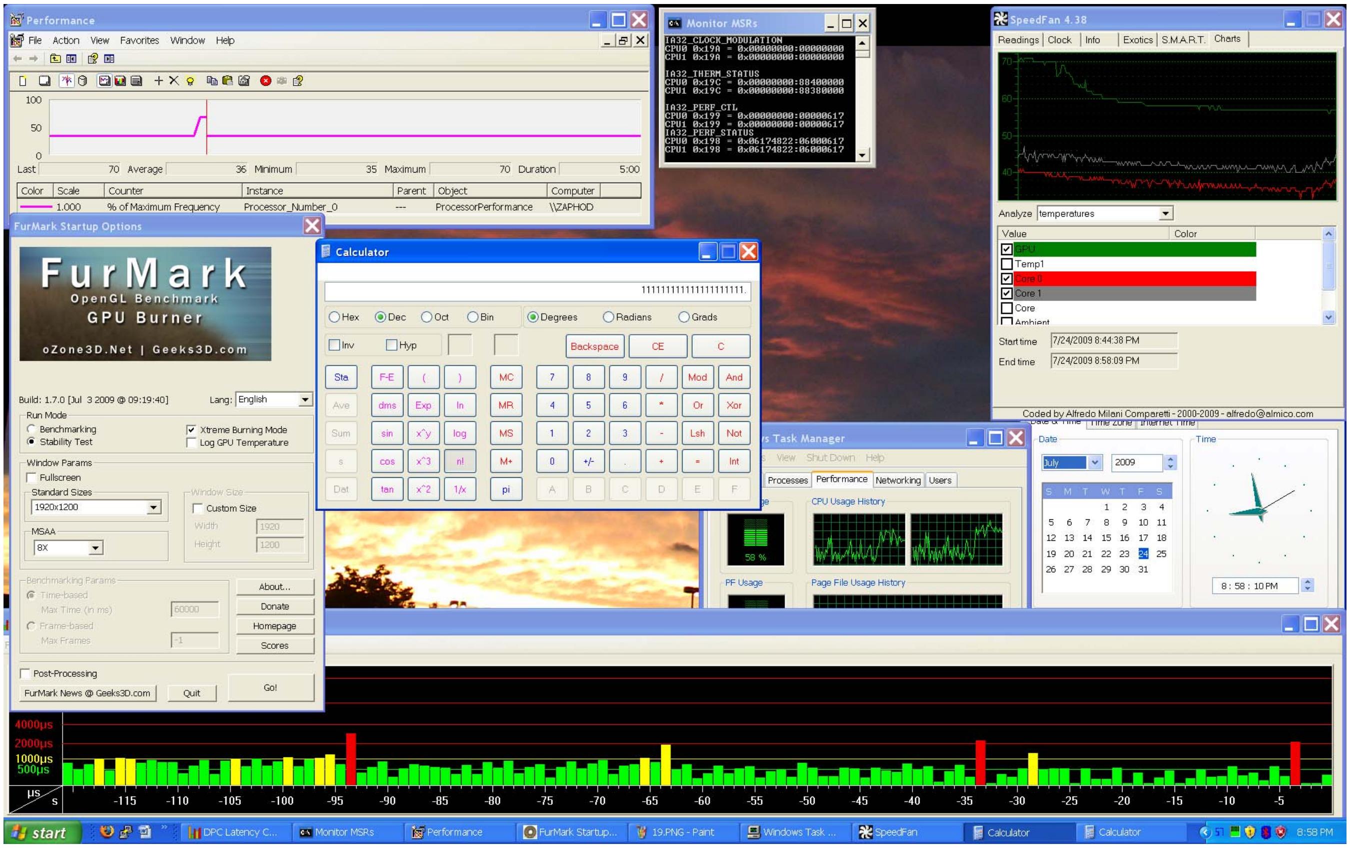
Now at 75% modulation.



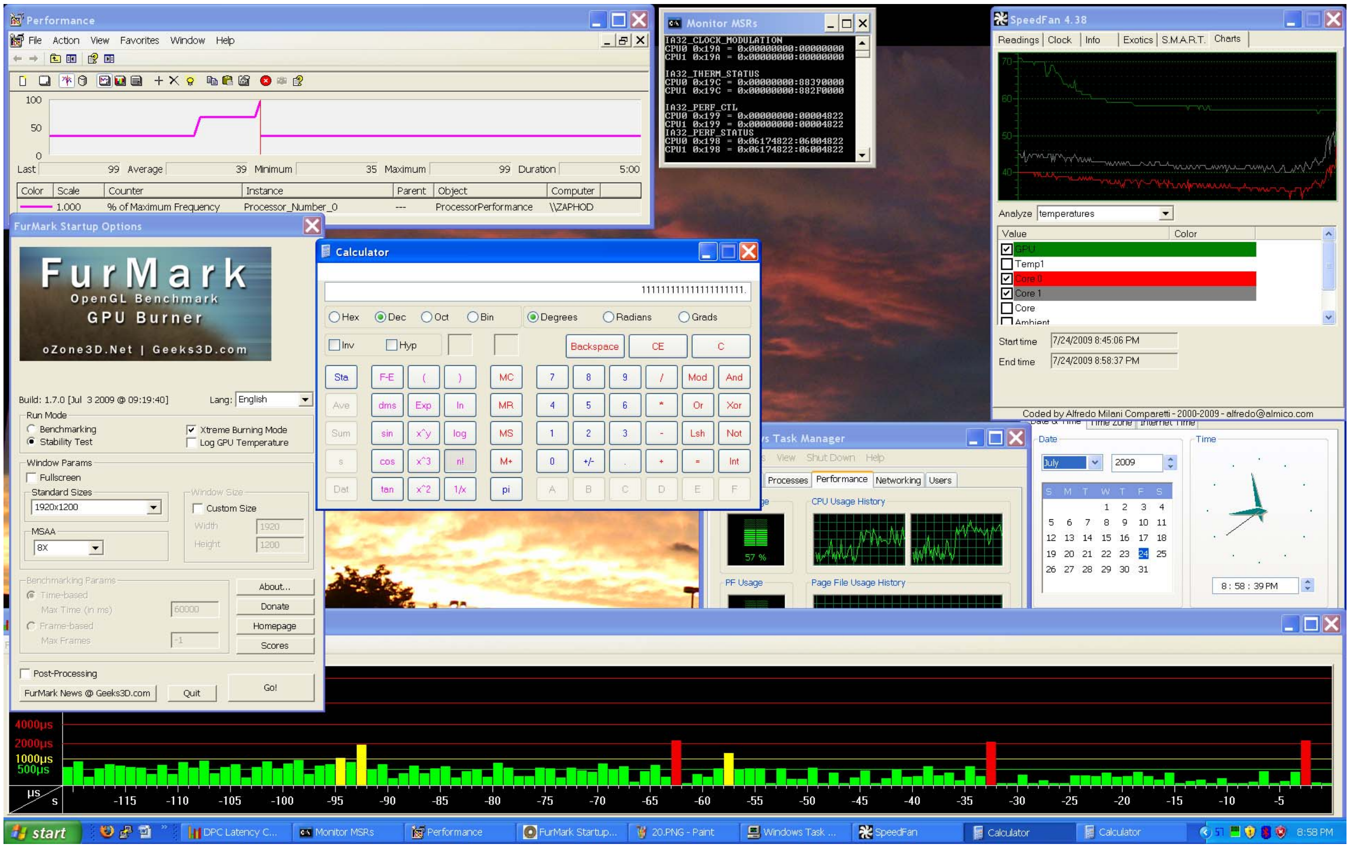
Now at 87.5% modulation. Note that a Windows Calculator window has been started (though no operation was initiated yet). To further demonstrate how little connection there is between the system's throttling and the CPU core temperatures, a long-term calculation process will be started to elevate CPU temperatures (factorial calculation of a large number). Screen captures will show that all the throttling measures will continue to be reversed even in the face of rising CPU core temperatures.



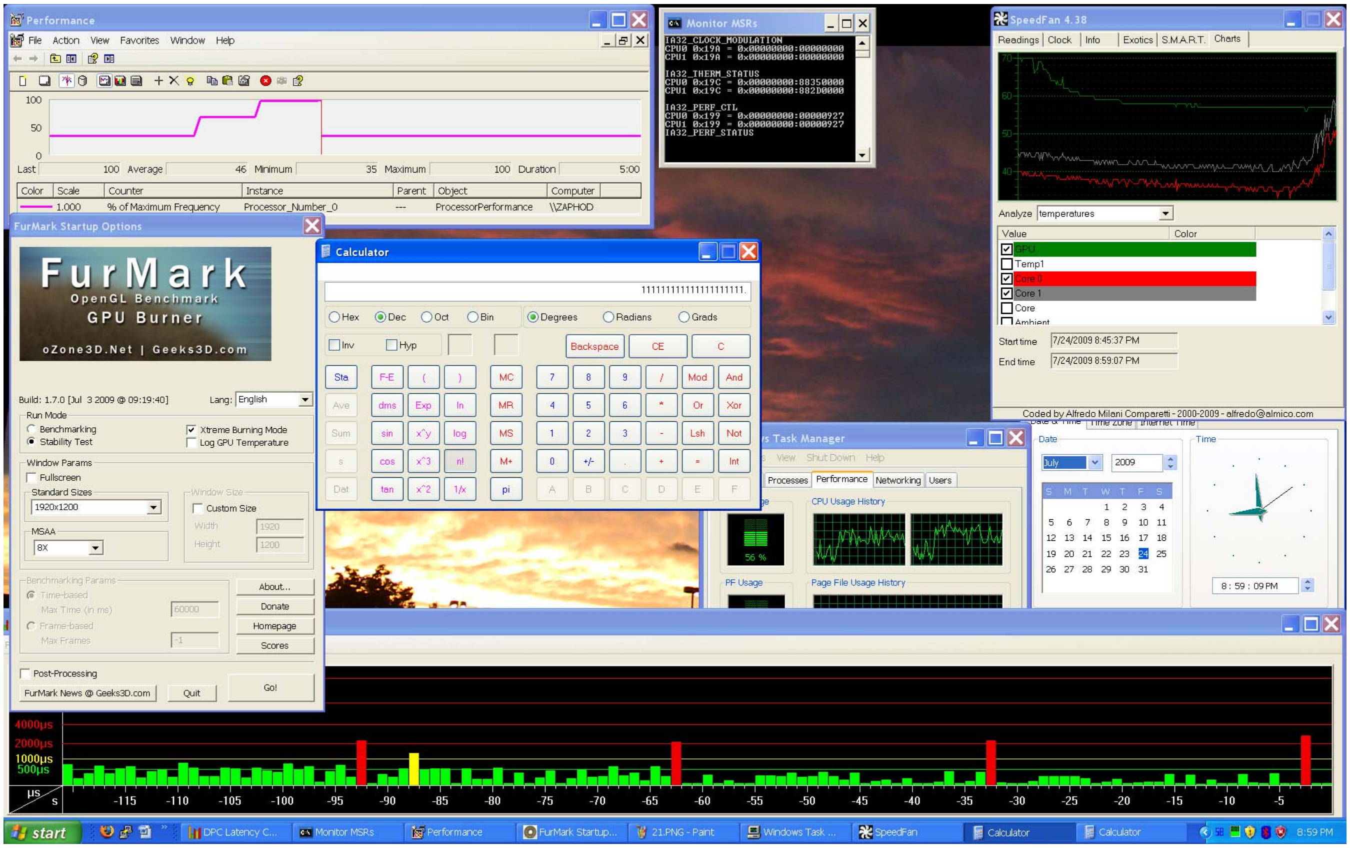
Software-controlled Clock Modulation is now turned off. Reversal of P-state transitions are next. The calculation of 111111111111111111111! is started in Windows Calculator.



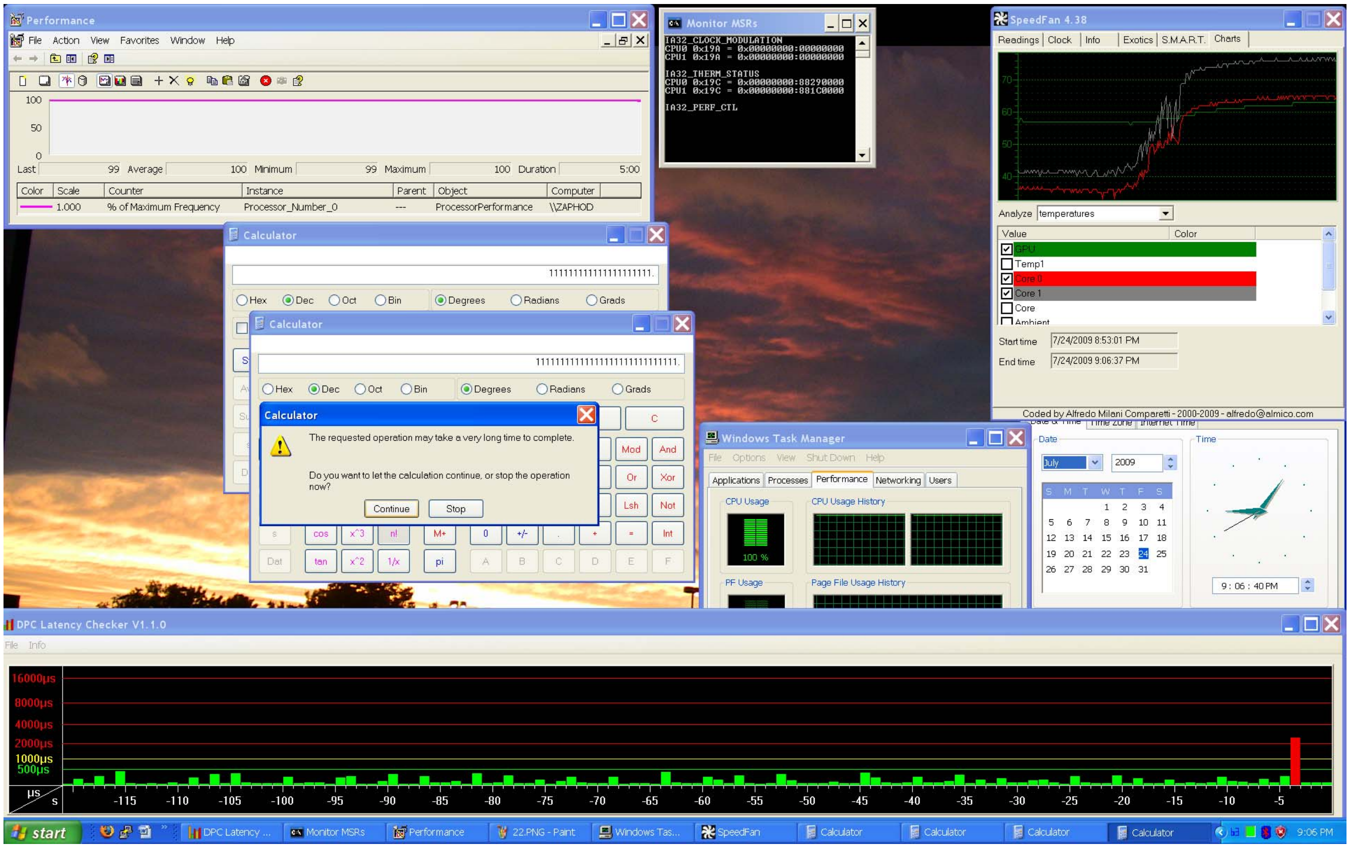
The transition from P3 to P2 is shown (Front Side Bus frequency is restored and CPU performance is now at 70% of total capacity). Note that CPU core temperatures are now on the rise.



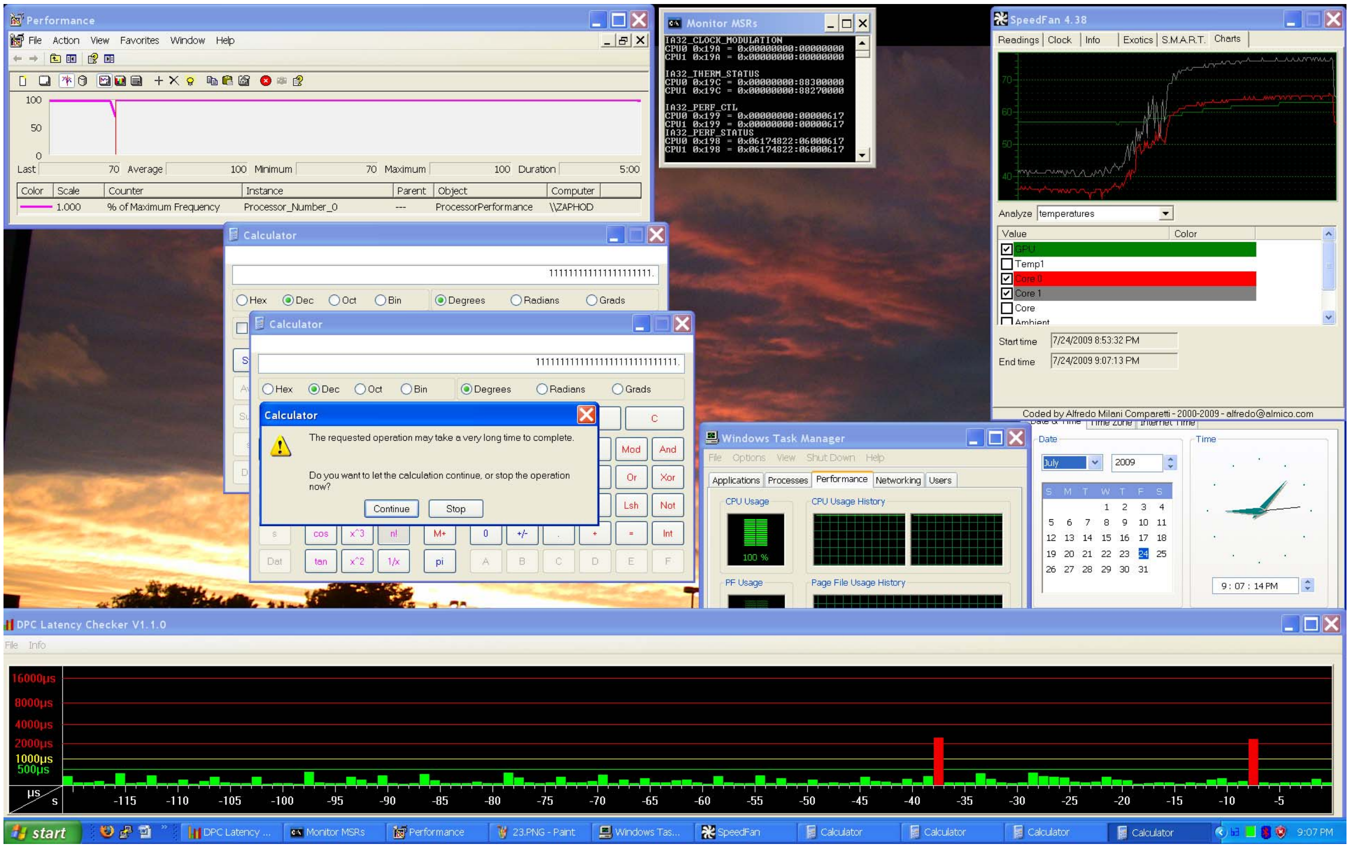
The transition from P2 to P1. This transition took place despite dramatically escalating CPU core temperatures, further demonstrating that there’s not much correlation between the throttling process and CPU temperatures.



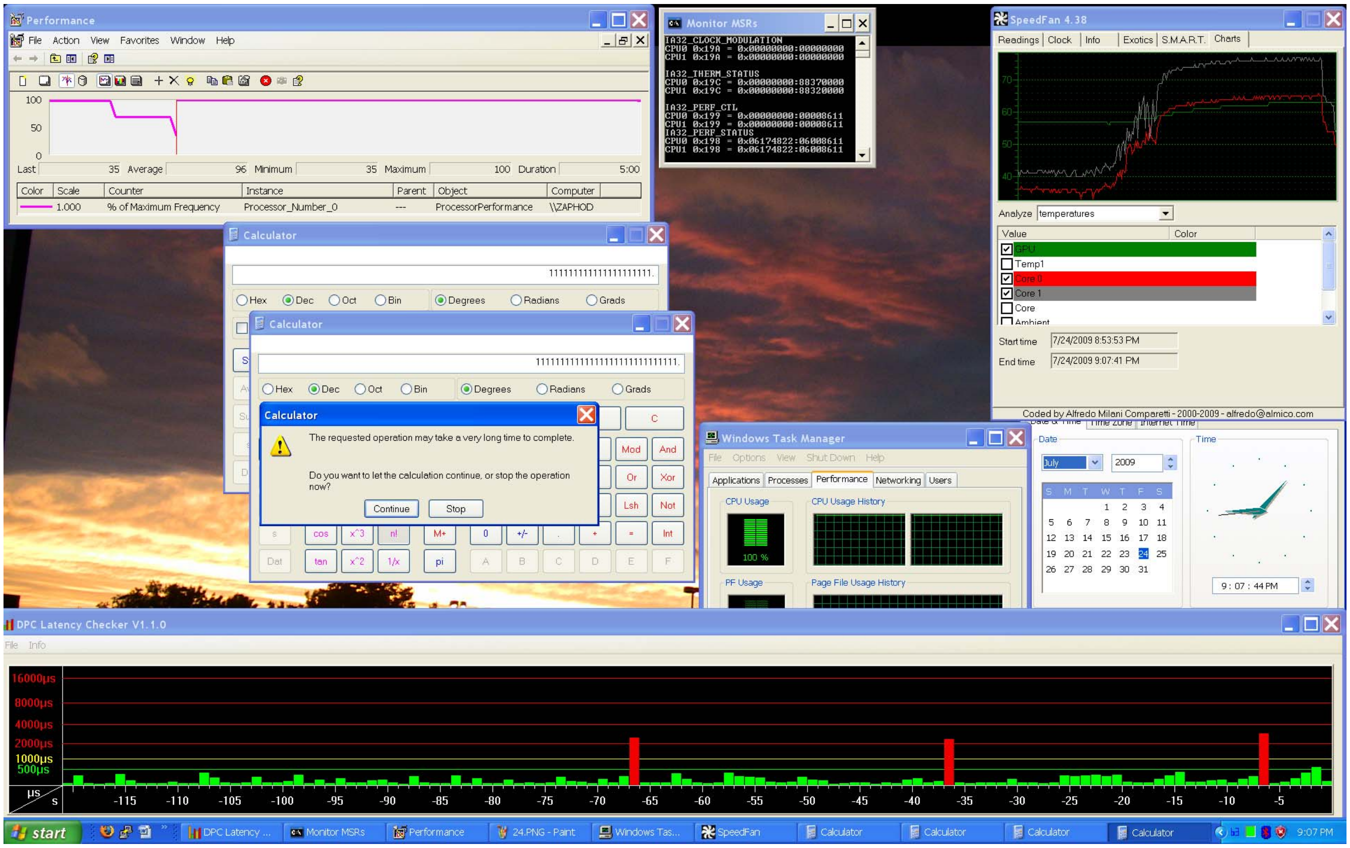
The final transition restoring the system to full processing capacity. Again note that core CPU temperatures are taking off, but this last dethrottling step still took place. However, while the correlation between GPU temperature and system throttling is much higher than for CPU temperatures, the next few screen captures demonstrate that there is nonetheless some finite CPU temperature correlation. At this point a second Calculator process is started, with the same long-term calculation, to bring both CPU cores to 100% utilization and escalate the CPU core temperatures.



About 7 minutes later, the GPU is at 63°C (warmed somewhat by the CPU, which is nearby on the motherboard), but not near the 69°C that was required to trigger throttling before. Yet throttling is indeed being triggered, with the first event having just occurred in the screen capture.



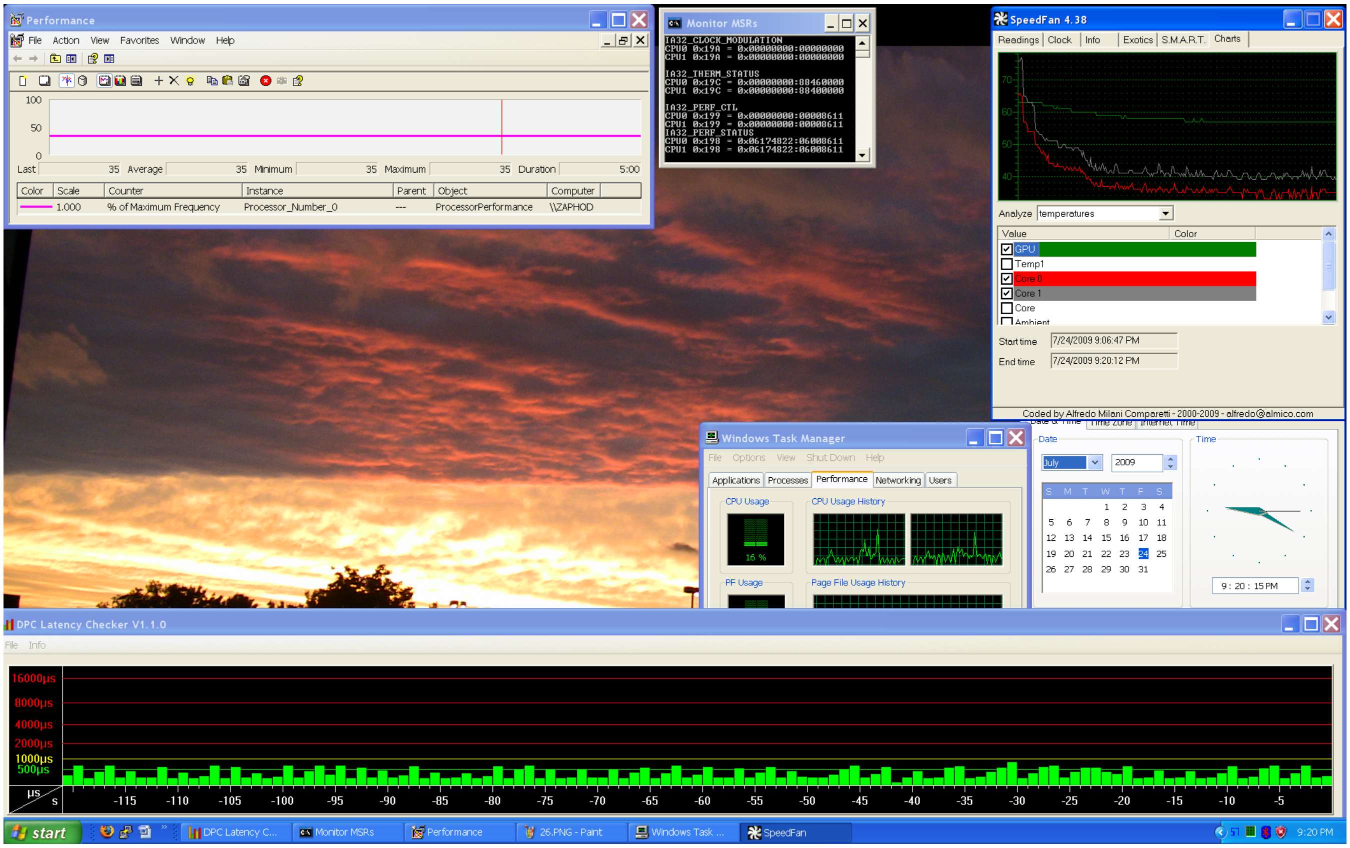
The next performance state transition takes place on schedule, as before, now down to 70% capacity.



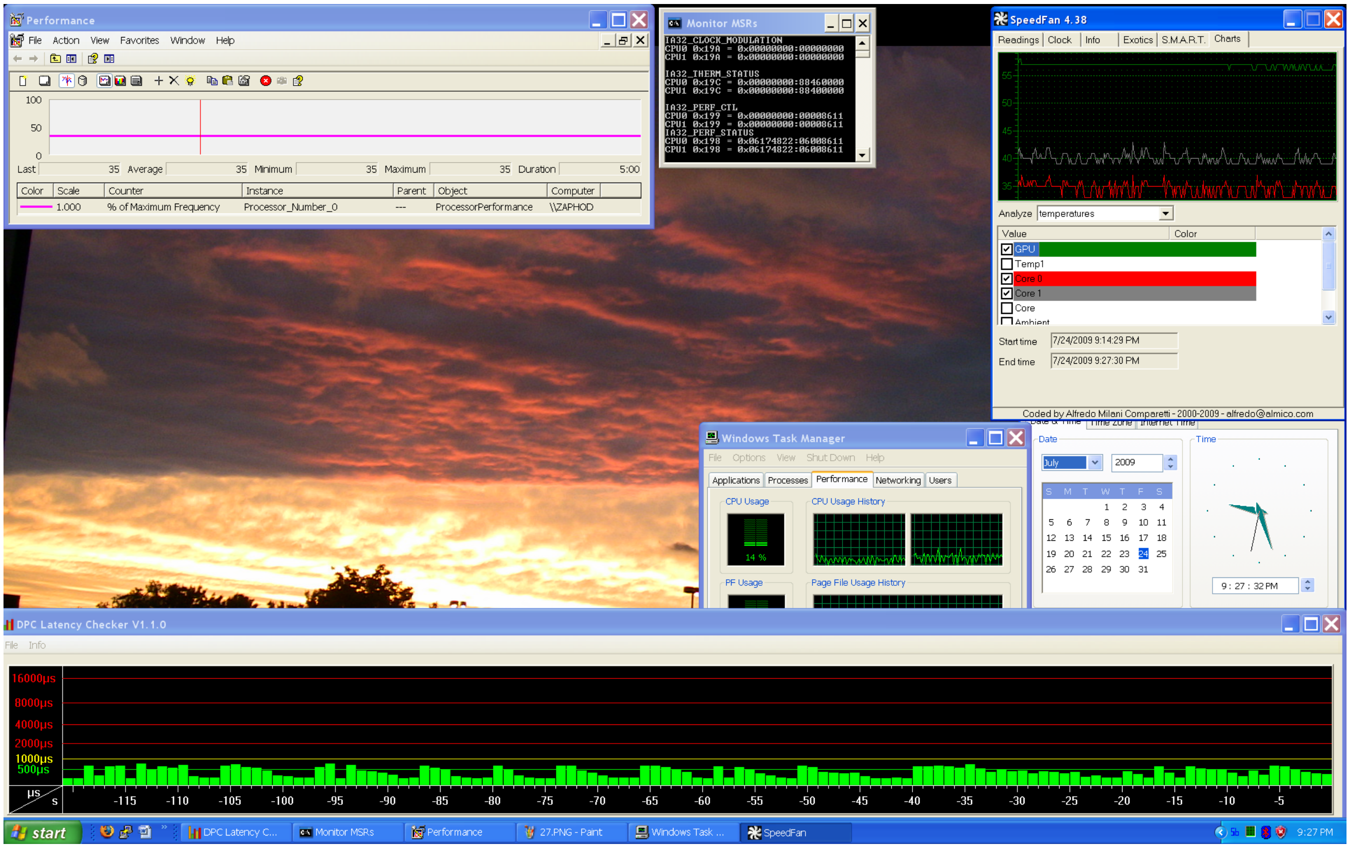
The transition to P3 and 35% capacity (along with halving of the FSB clock frequency)

But then there are no more transitions. That is, throttling stopped with the transition to P3 (which is shown on the far left of DPC Latency Checker's window, having taken place about 2 minutes prior). It would appear that the polling process took the dramatic drop in CPU temperatures into account and decided no further throttling was necessary. So it appears that CPU temperatures do have some effect, though this test and later tests show that GPU temperature determines, much more than the CPU, whether and when throttling takes place.

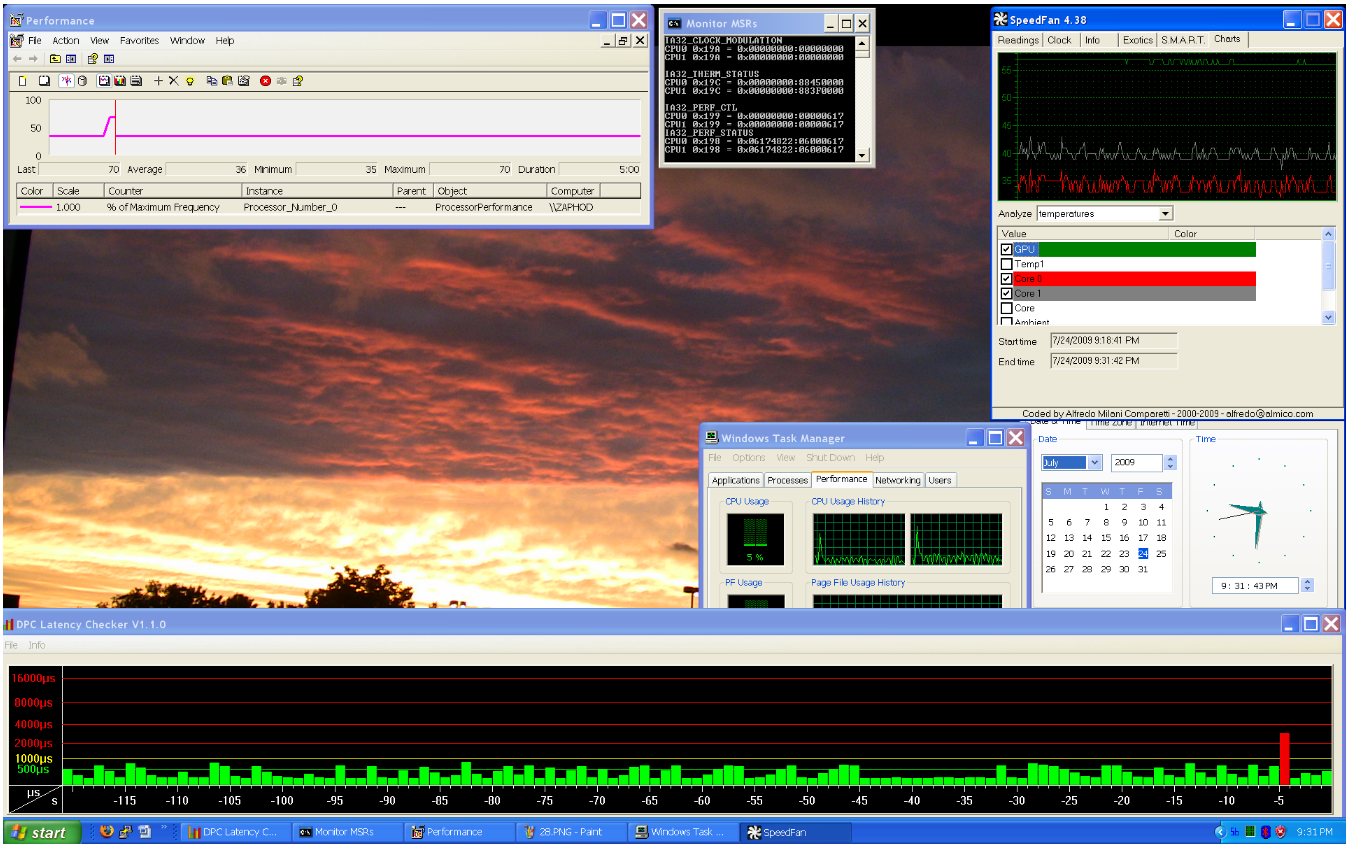
To continue this test, the calculator processes have been stopped just prior to this capture in order to let the system cool down and see how and when throttling is reversed.



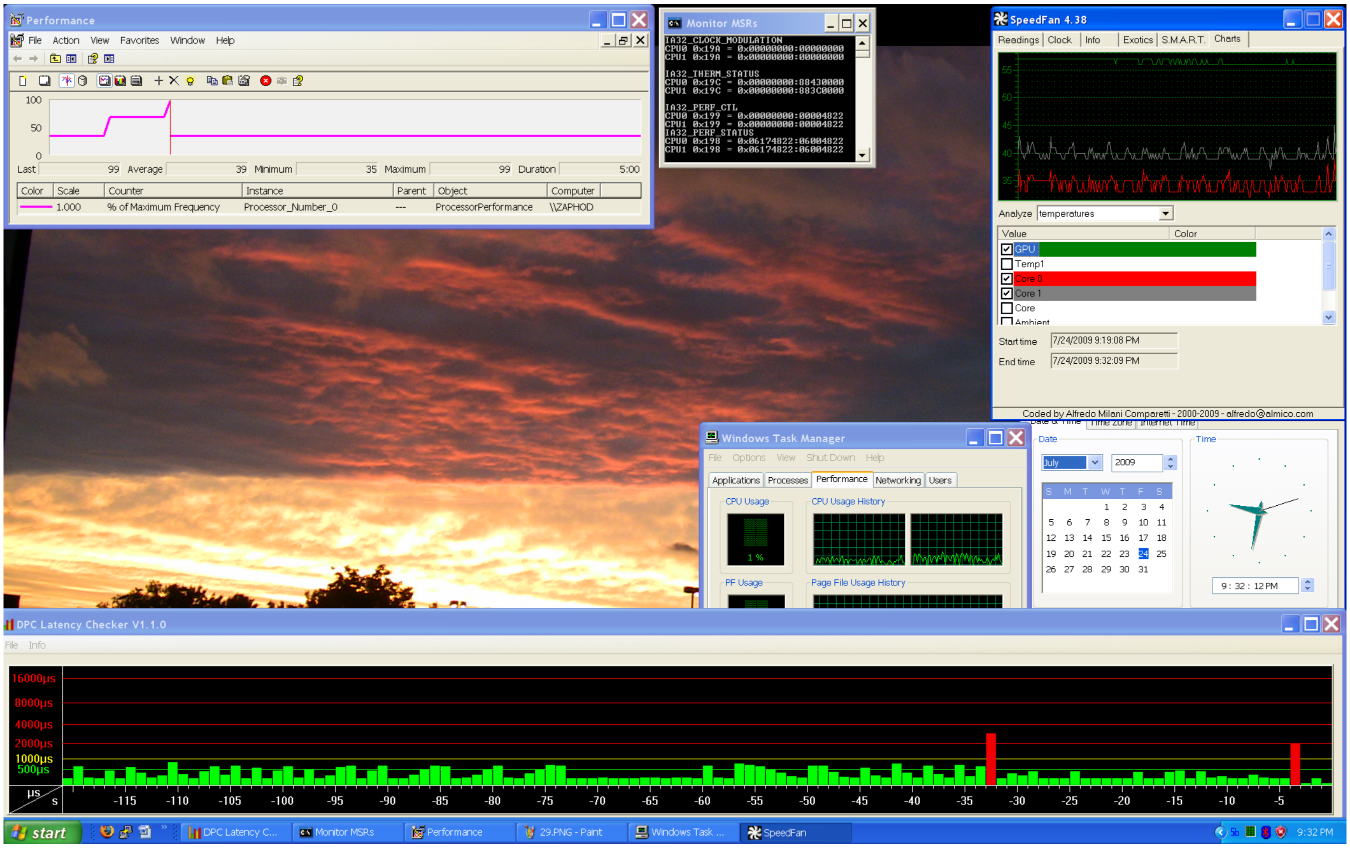
More than 10 minutes later, however, despite temperatures having returned to normal idle levels, performance is still substantially throttled, remaining at 35% with a half-speed Front Side Bus.



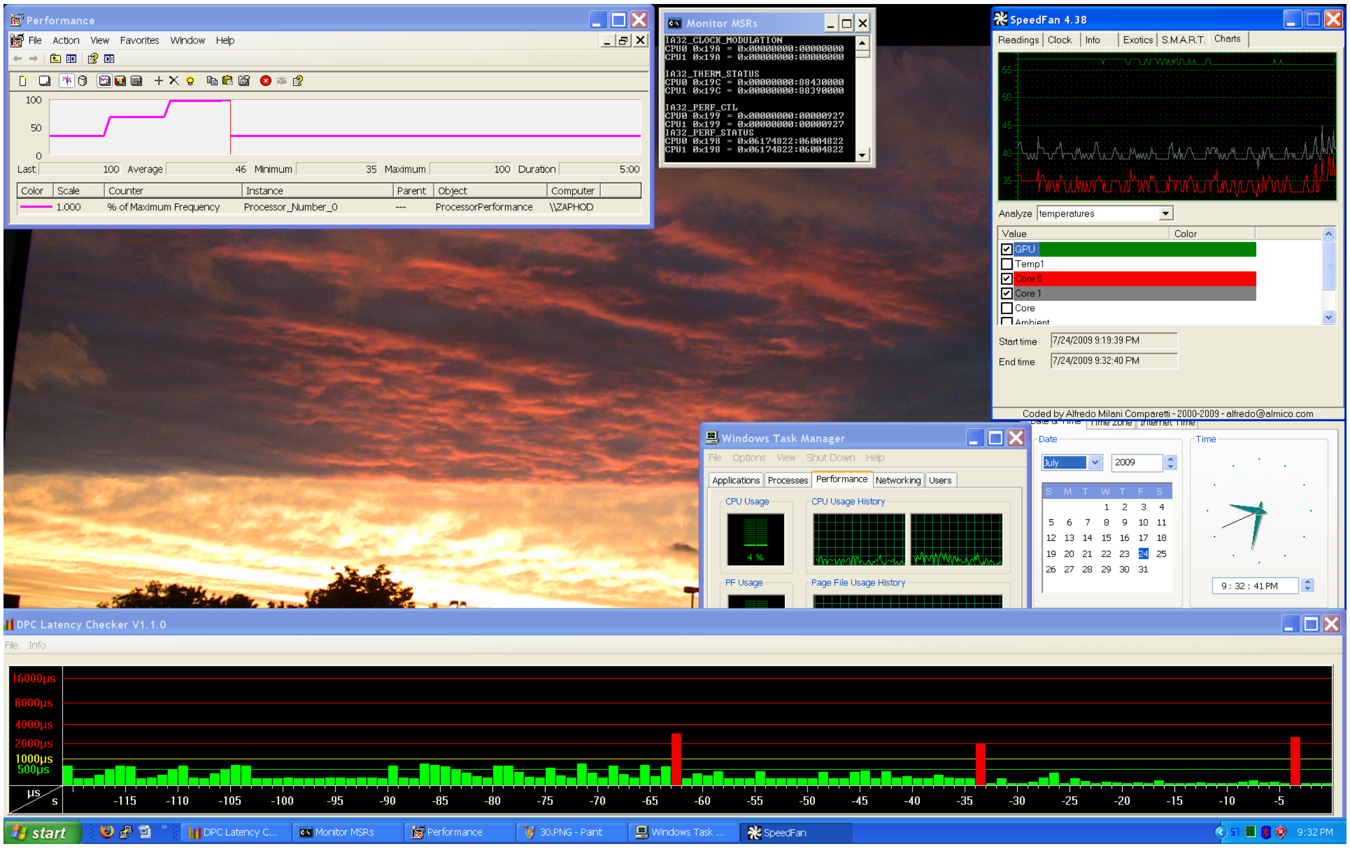
Now 18 minutes since the calculator processes were stopped, but throttling is still in place.



More than 20 minutes after the system was left idle, throttling was still in effect. Moments before this screen capture above, a small desk fan was aimed toward (but not directly at) the right side of the docked E6500 to create a gentle circulation of air in an attempt to nudge the system into reversing the throttling. That effort was successful as the transition to state P2 from P3 did finally take place as shown above.



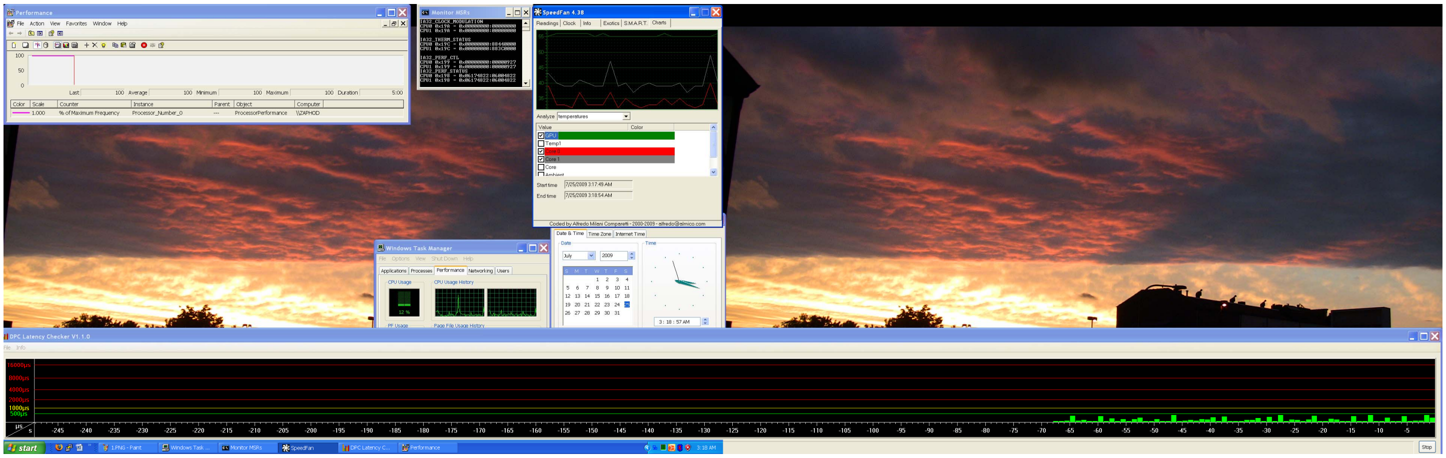
30 seconds later, right on schedule, the transition from state P2 to P1.



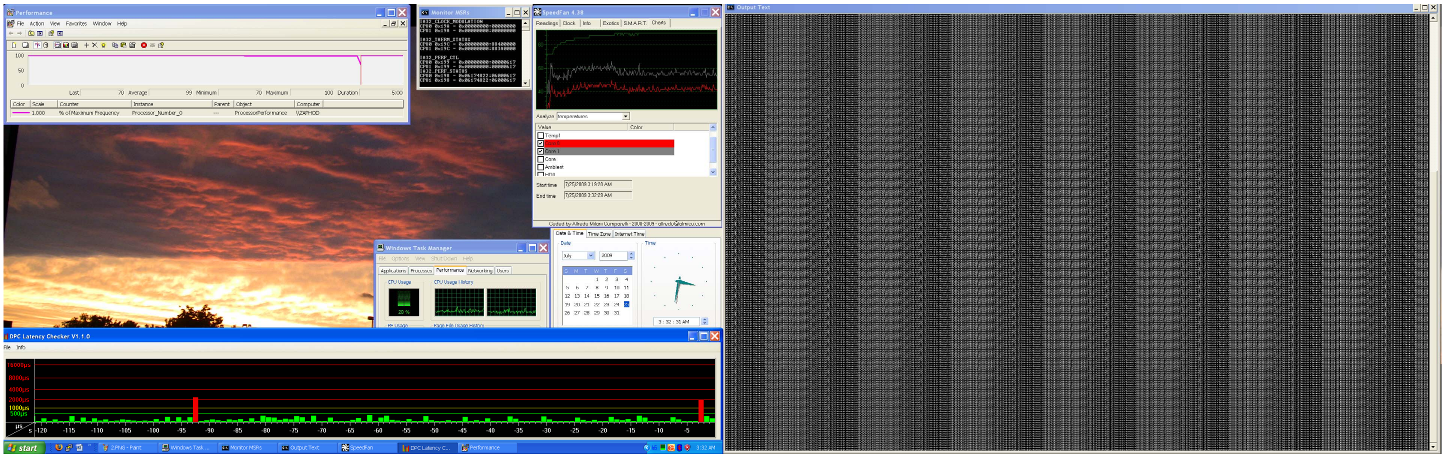
Finally, the last throttling step is reversed (with the continuing influence of the external desk fan) and the system is back to full processing capacity. That completes this test session.

Illustration: throttling can engage without the aid of special processor-stressing utilities

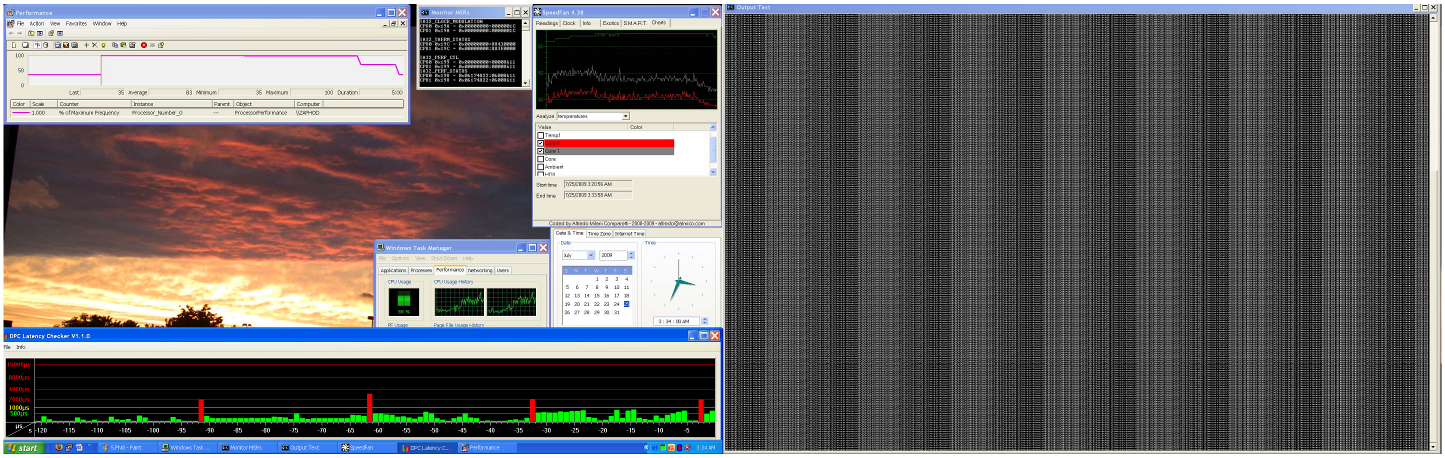
Specialized processor-stressing utilities are not necessary to evoke system throttling. In the following test, a single simple batch file process (revealed in Appendix B) is used to fill a screen with scrolling text. Throttling is invoked 12 minutes later with the GPU temperature only at 65°C and CPU core temperatures both under 50°C. In this test, the Dell Latitude E6500 is docked to the Dell E/Port Plus Advanced Port Replicator, as in the last test. Dual monitors connected to the E/Port Plus are used, though, instead of a single monitor. Testing has shown that the throttling effects are somewhat worse when dual monitors are in use. Since throttling is closely correlated with GPU temperature, this is something one would expect as dual monitor operation would place a greater demand on the GPU. The ambient room temperature for this test was 27.5°C.



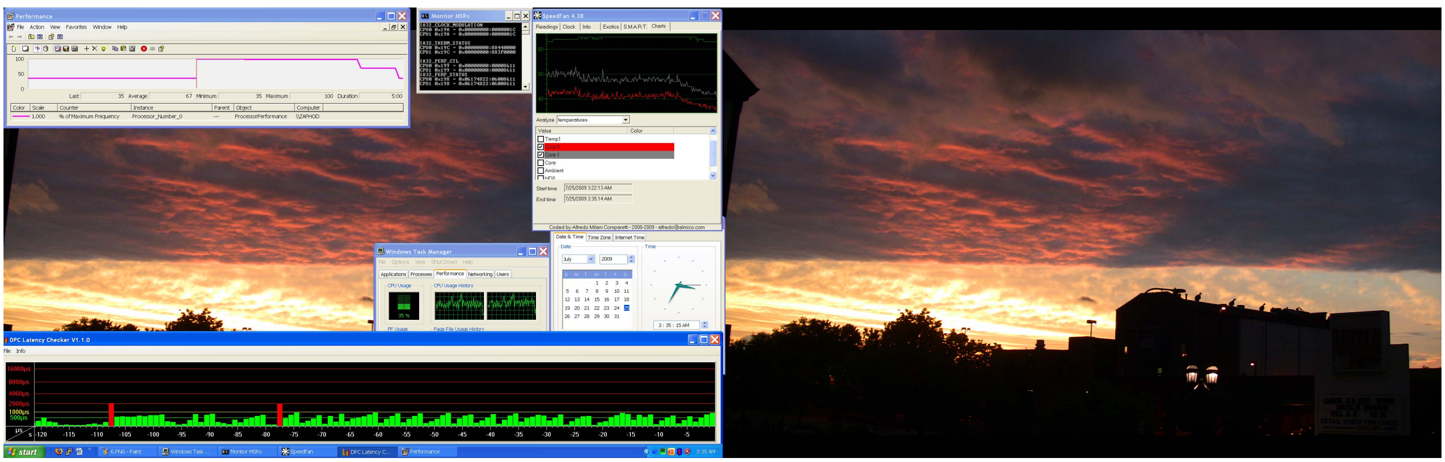
This first screen capture above, taken just before testing started, shows no throttling in effect yet, GPU temperature around 55°C, CPU core temperatures averaging below 40°C, CPU latency under 500 µS and low processor utilization.



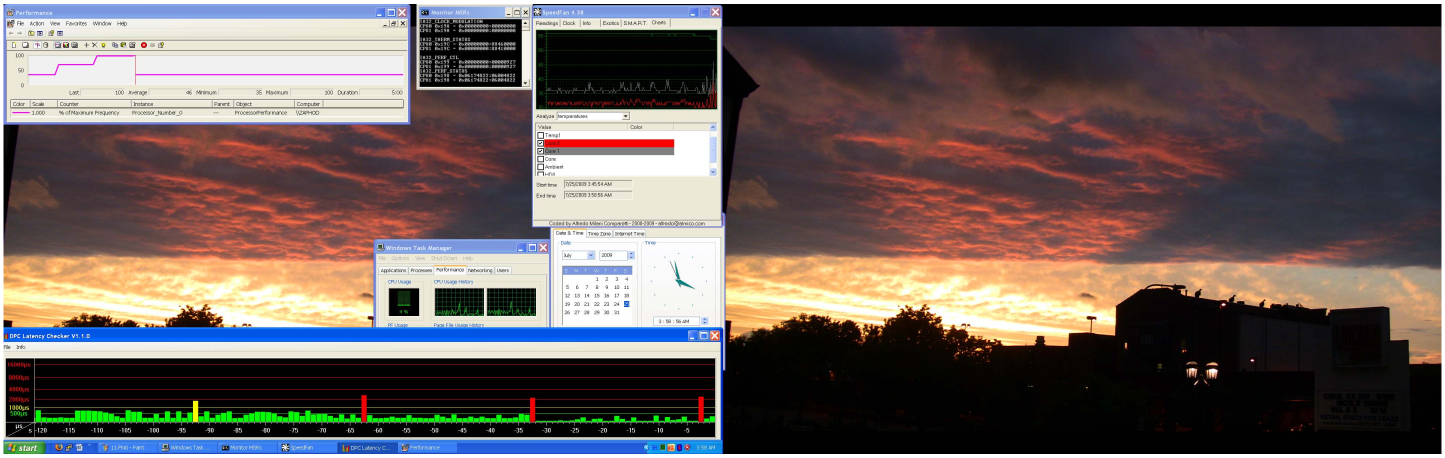
At this point, about 13 ½ minutes into the test, the first two throttling events have occurred, even though GPU temperature is only at 65°C and the CPU cores remain under 50°C. The first two events shown didn't take place 30 seconds apart, but they were still separated by 30 second intervals (90 seconds apart).



At this point, one and a half minutes later, 3 additional throttling events have taken place. The system is now degraded by Software-controlled Clock Modulation to 75% capacity of performance state P3, for an equivalent processing power of about 600MHz.



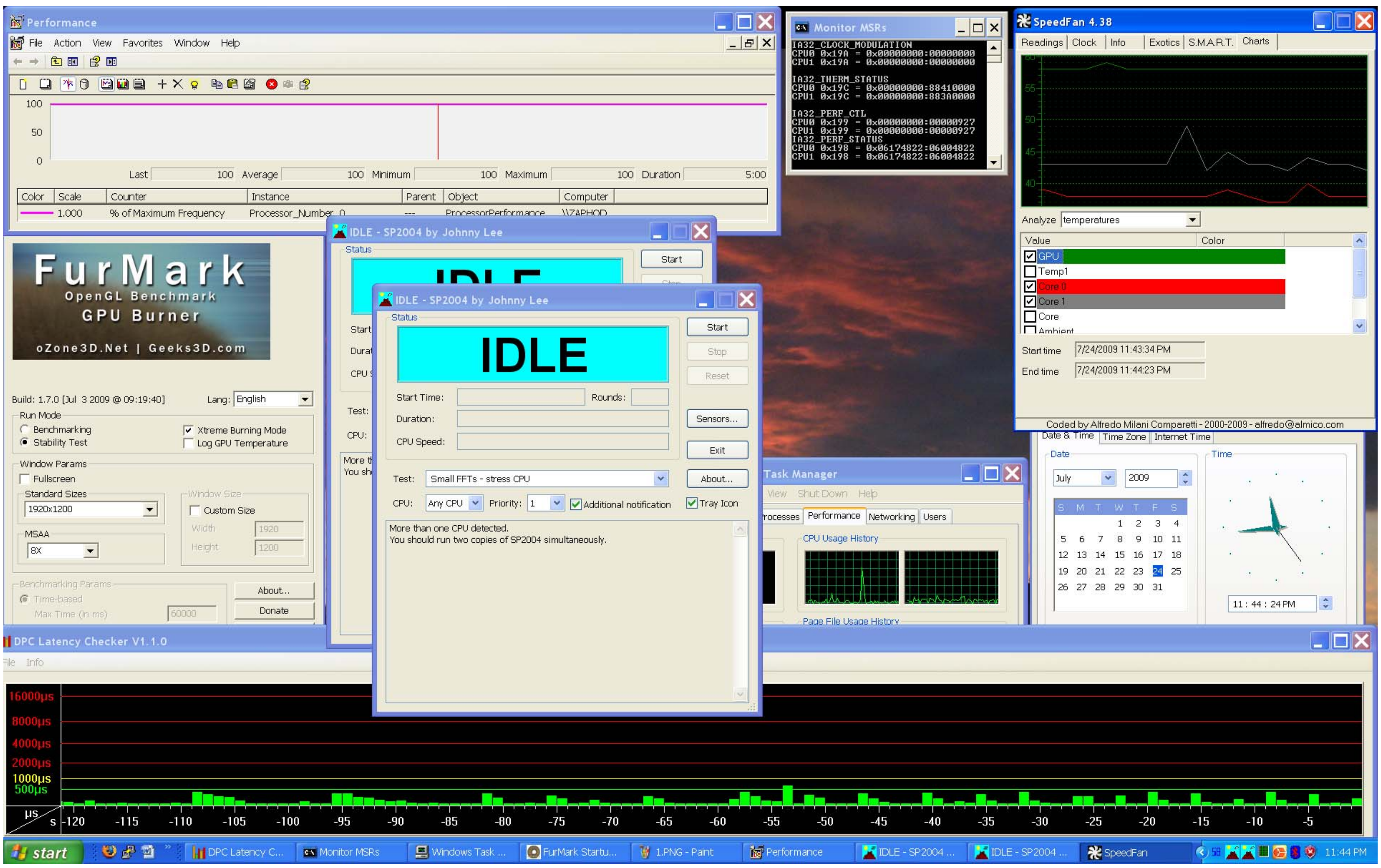
About one minute and 15 seconds later, it becomes apparent that no additional throttling will likely take place. Both GPU and CPU temperatures are falling and Software-controlled Clock Modulation is holding at 75%. The text output batch file was ended just before this capture to allow for cooling so that throttling reversal can be observed.



More than 20 minute later, throttling reversal finally began. The above screen capture shows the final four dethrottling events, restoring the system to full processing capacity.

Illustration: throttling occurs at higher temperatures when not docked

The effects of throttling are somewhat reduced when the system is operated without being docked to the E/Port Plus Advanced Port Replicator. Higher temperatures are required to trigger throttling events (on the order of about 10°C higher). These higher temperatures are still well within the normal operating range of the CPU and GPU, however. In the following test, both the CPU and GPU are stressed at the same time using separate utilities (Stress Prime 2004 and FurMark) until throttling is triggered at temperatures around 80°C. The system performance becomes degraded by all 10 throttling steps after about 17 minutes and about 8 minutes are required to reverse the throttling after the stressing utilities are stopped. The system is not docked, it is plugged in to AC power and a Dell E248WFP is used to mirror the image on the laptop's screen (via the rear DisplayPort). In other testing, very similar results were produced for the undocked configuration whether or not AC power was present and whether or not an external monitor was attached.



The screen capture above was taken just before testing began. Both FurMark and Stress Prime 2004 will be used to exercise the system since prior testing showed that neither utility alone was sufficient to trigger throttling. The ambient room temperature for this test was about 28°C.

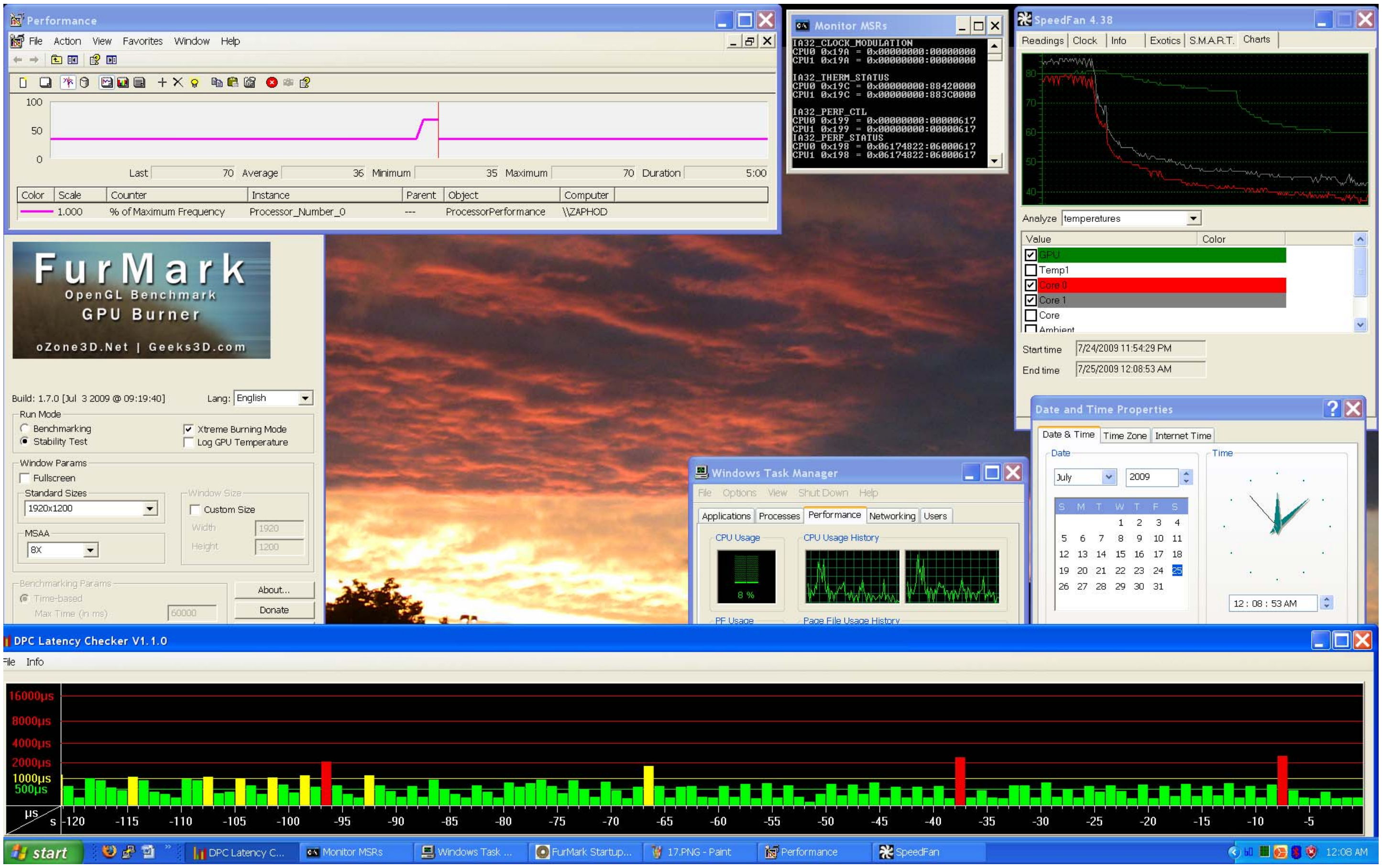
About 12 minutes later, the first throttling event occurs. The image in the background is the FurMark rendering window. GPU temperature is just over 80°C, about 10°C higher than when throttling was first triggered in the docked single monitor configuration test and about 15°C higher than in the docked dual monitor configuration test. 80°C is still well within CPU and GPU designed operating range.

Here we can see how DPC Latency Checker recorded the occurrence of the first 7 throttling events. The remaining three throttling events are shown in the next capture.

The remaining 3 throttling steps are shown to have occurred here, leaving the system fully throttled (12.5% of the P3 State, or about 100MHz overall, compared to its 2261MHz capacity). Both the FurMark and Stress Prime utilities were stopped just after this capture.

About 4 minutes later, upclocking begins and proceeds as usual, progressively reversing the throttling steps at 30 second intervals.

Shown here is the status after 5 upclocking events. 5 more events remain.



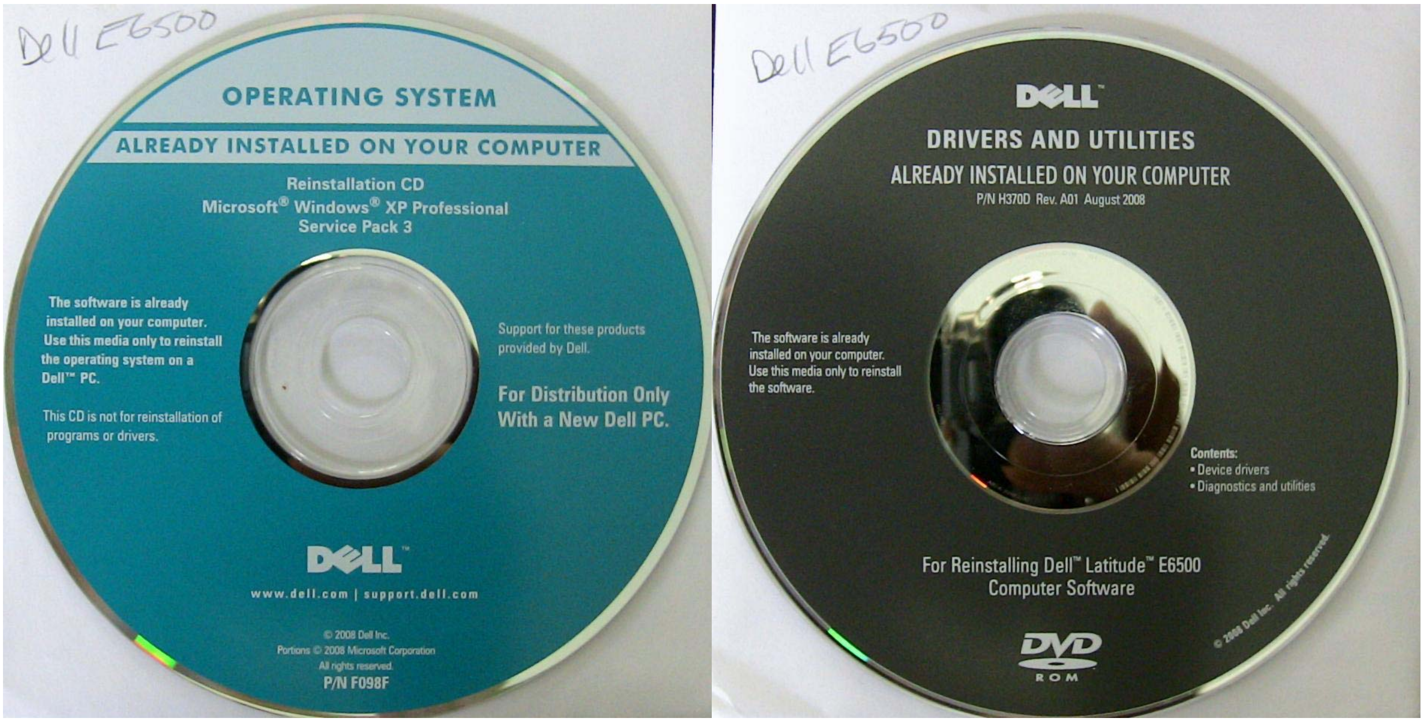
Here is the status after 3 more upclocking events.

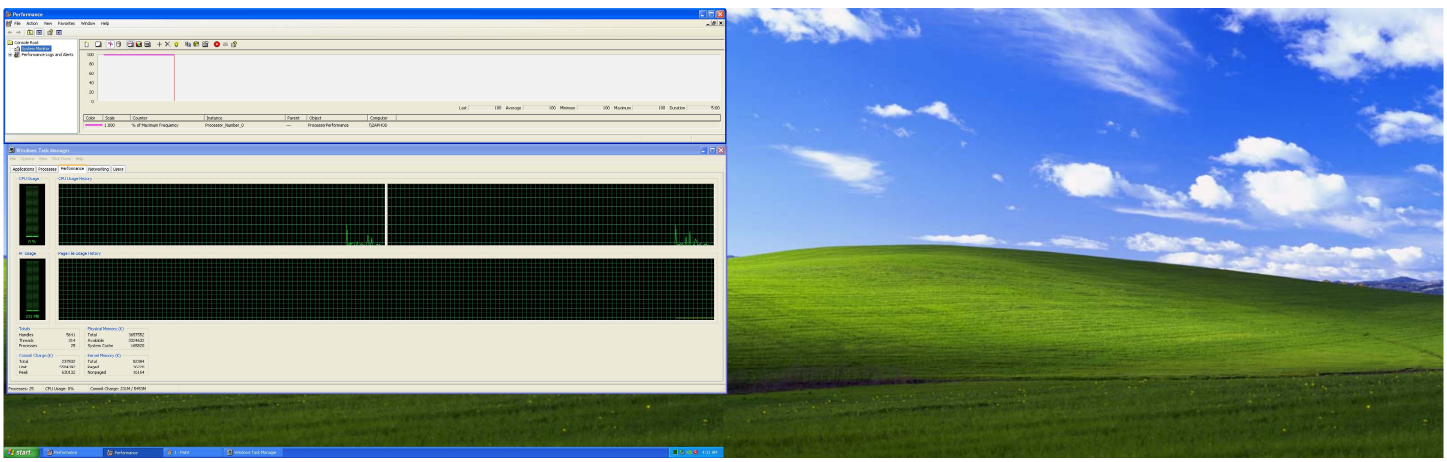
Finally, the two remaining upclocking events finish out the throttling reversal and the system is back at full processing capacity. The only meaningful observed difference in the throttling behavior in a standalone configuration as opposed to the docked situation is that the throttling process starts (and reverses itself) at higher temperatures.

Illustration: throttling occurs even with no third party software present

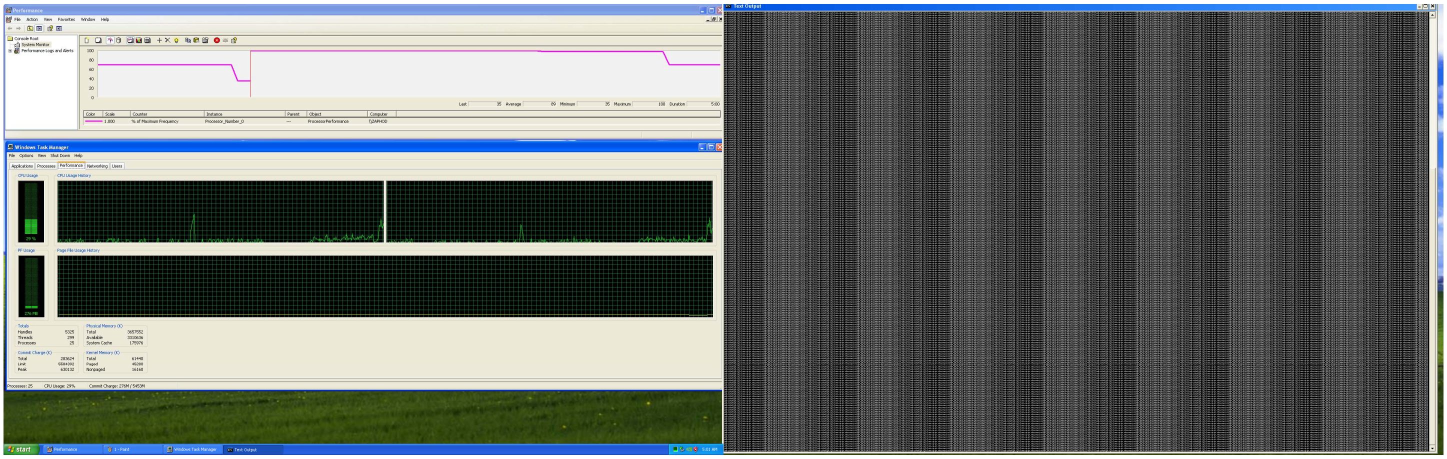
One problem that troubleshooting this phenomenon presents is that generally those who are affected by it already have installed many third party software packages on their system. One can make the argument that the throttling behavior might not have been present on the system as received from the factory, but was rather introduced as the result of third party software. In this test, the Dell Latitude E6500, is reconfigured with only Windows XP Service Pack 3 and the Dell-supplied NVIDIA video driver. In addition, no third party software is installed to conduct the test. Only the standard Windows XP utilities perfmon.exe and Task Manager are used to monitor the system and the throttling is triggered by the same simple 4-line looping batch file used in a prior test to generate scrolling lines of text (see Appendix B).

In this test, the entire original drive shipped with the Dell Latitude E6500 was reformatted, destroying all data, in favor of a fresh installation of Windows XP Service Pack 3, using the original Dell-included installation CD. Then, the NVIDIA Quadro NVS 160M driver was installed from the Dell-included “Drivers and Utilities” DVD-ROM. Below are photo images of the CD and DVD-ROM used.

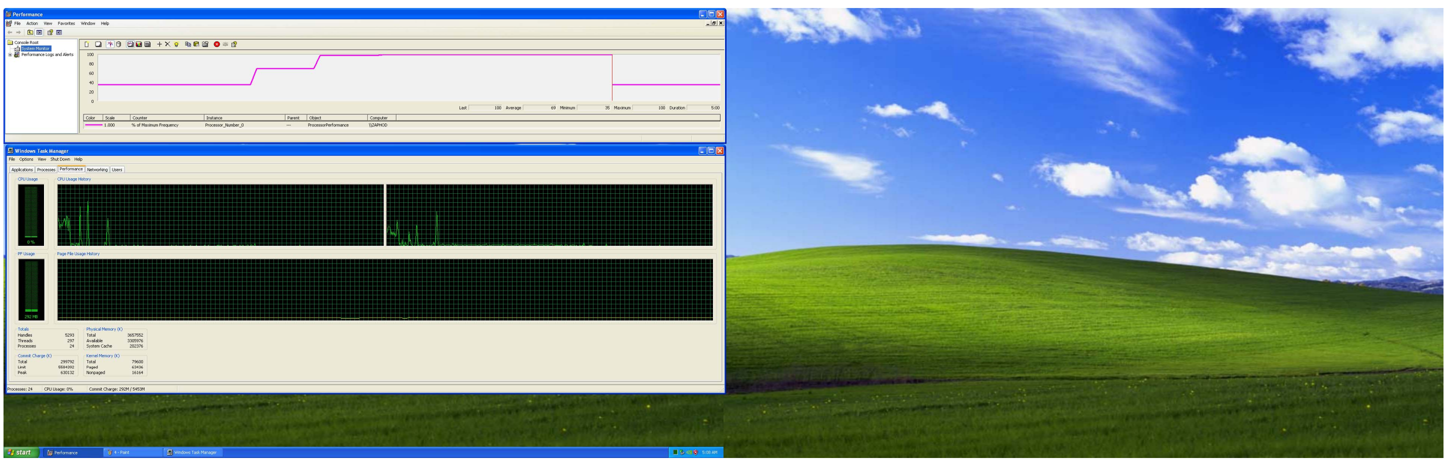




The screen capture above was taken just before testing. The system was docked in dual-monitor mode for this test. Ambient room temperature was 28°C.



Though it took longer to trigger throttling in this case, half an hour later the system was throttled down to at least 35% of its true capacity. It wasn't possible to measure whether Software-controlled Clock Modulation was engaged since third-party software was not a part of this test by design. However, it stands to reason that this is the same throttling mechanism observed in prior tests, particularly since the throttling events occurred at the usual 30 second intervals (the P1→P2 and P2→P3 transitions were separated by 60 and 90 seconds, respectively) . The batch file was stopped immediately after the screen capture above.



About 7 minutes later, the throttling was reversed in the expected 30-second interval steps.

<u>Conclusions:</u>

This article provides ample evidence to suggest that the Dell Latitude E6500 was shipped to customers with a defect that can dramatically reduce the system's performance capacity by more than 95% under normal operating conditions, especially when docked to the E/Port Plus Advanced Port Replicator. This phenomenon has been identified in Internet forums since late 2008. Similar complaints have also been made regarding the Dell Latitude E6400. Complaints have been lodged regarding both possible graphics configurations (integrated Intel graphics and discrete NVIDIA Quadro NVS 160M graphics) for both the E6400 and E6500. A particularly unfortunate aspect of this apparent defect is that it's unlikely it would be properly diagnosed by end users and perhaps even some technical professionals. Users may suffer from the adverse effects of this apparent defect for months or years since there is no outward indication of the cause. The symptoms only manifest themselves as general sluggishness in degrees varying from annoying to unusable, occurring at what may seem like random times to most end users. Even technical support personnel are perhaps more likely to presume that Windows or

application software or network problems are to blame, yet find themselves unable to pin down the software or network issue to blame because the cause lies elsewhere.

As dramatic as the above test results are, all test were conducted at an ambient temperature of 28.5°C or less. Yet the Dell Latitude E6500 specifications define an operating temperature up to 35°C[12]. E6400 specifications show an operating temperature up to 40°C[13]. At such higher temperatures, throttling will undoubtedly occur sooner and be more dramatic. And, perhaps most significantly, it will be more likely that throttling will not reverse for a very long time once it is engaged (as happened in some of the tests above) and perhaps not even at all.

Possible remedies:

It seems clear that some error or oversight occurred on Dell's part regarding this system's design. Even if the argument is made that this aggressive processor throttling was deliberately designed to keep the system cool, this would not explain why the throttling is more aggressive when the system is docked. A docked system is not sitting on a user's lap, so discomfort from heat is much less of a concern. In addition, a docked system is much better ventilated due to the additional clearance provided for the grating on the bottom of the chassis, which constitutes the primary air inlet. It stands to reason that the better ventilation provided by the stable platform and the increased clearance should actually allow for safe operation at somewhat *higher* processor temperatures.

The quickest, easiest stopgap remedy might be to release a software patch (or a new BIOS flash, if necessary) that configures docked system throttling in exactly the same way as undocked throttling. That alone would be a major improvement, since undocked throttling seems to occur at temperatures 10-15°C higher than in docked systems.

Beyond that, assuming for the moment that the throttling is taking place in an ACPI-compliant manner under control of Windows, there could be a number of ways to rectify this problem with a new BIOS release. The following adjustments might be made:

1. The _PSV threshold for engaging passive cooling could be raised substantially, especially for the docked configuration.
2. The lower setting for _PSV used for hysteresis once passive throttling is engaged could also be raised substantially in a corresponding manner.
3. The _TSP value for passive cooling polling period could be shortened significantly. Thirty seconds seems much too long to wait for making these adjustments, especially if they're made only one step at a time.
4. The _TC1 and _TC2 constants, if used, could be adjusted for faster convergence to the lower (hysteresis) setting of _PSV.
5. The throttling algorithm could be much less aggressive overall and could eliminate perhaps even the majority of the most severe throttling points. It doesn't seem necessary to cripple the system down to a few hundred MHz unless temperatures are truly near a critical threshold in the neighborhood of 100°C.
6. It appears these changes could be made with ACPI name definitions and control methods in a new BIOS release (as part of a DSDT or SSDT block).

There are two obvious, larger, overarching questions as well:

1. Is this throttling even necessary at all? The Core 2 Duo processor has built-in throttling once temperatures really do reach a critical stage (105°C at the cores) and should also shut the system down if temperatures rise to levels where damage can occur. Do the Intel and NVIDIA GPU's not have the same features? Isn't that enough?
2. If this throttling really is necessary for some reason, shouldn't the user
   A. Have some control over how it is configured, or at least…

B. Receive clear and obvious notification when it is engaged?

Finally, since many users may not be aware that they are being adversely affected by this apparent defect, it seems important that Dell directly communicate the problem and directly offer the remedy to users of the affected systems. In fact, perhaps the most appropriate and commendable approach would be a proactive, formal and widely publicized recall. Even if the problem can be fixed with a BIOS flash performed by the user, the issue is serious enough and at the same time so insidiously invisible that it should be publicized widely to ensure as much as possible that all affected users learn of the problem and its solution. To preserve their reputation as much as possible, perhaps Dell should offer to perform whatever fix is necessary free of charge (including shipping), regardless of warranty status, since this appears to be a fundamental and highly consequential design flaw.

Disclaimer:

I, as the author of this article, do not work for any corporation. I was not compensated in any way to write this article or perform the research used in writing it. I am an independent software developer with some hardware background who was adversely affected by this apparent defect. I am no expert, but I know enough to troubleshoot, investigate and draw logical conclusions regarding computer and network problems. I am publishing this article to educate the public about this apparent defect and hopefully influence those responsible to provide a remedy to owners of these apparently defective systems.

Appendix A

A batch file to monitor MSR values

This command batch file simply calls the msr.exe command from the “Performance Inspector” package with the correct arguments to display the contents of various MSRs (Model-Specific Registers) in the Intel Core 2 Duo processor. The parsing feature of the “for /f” statement in Windows XP batch files is used to format the output of msr.exe. The “ping” command at the end of the loop is simply a way to insert a one-second delay.

```
@echo off
c:
cd \ibmperf\bin
:start
cls
echo IA32_CLOCK_MODULATION
for /f "usebackq skip=2 tokens=1,3-5" %%i in (`msr -r 0x19a`) do @echo %%i %%j %%k %%l
echo.
echo IA32_THERM_STATUS
for /f "usebackq skip=2 tokens=1,3-5" %%i in (`msr -r 0x19c`) do @echo %%i %%j %%k %%l
echo.
echo IA32_PERF_CTL
for /f "usebackq skip=2 tokens=1,3-5" %%i in (`msr -r 0x199`) do @echo %%i %%j %%k %%l
echo IA32_PERF_STATUS
for /f "usebackq skip=2 tokens=1,3-5" %%i in (`msr -r 0x198`) do @echo %%i %%j %%k %%l
ping –n 1 127.0.0.1 > NUL 2>&1
goto start:
```

Appendix B

A batch file to output scrolling lines of text.
Note that the blue text is all one line.

```
@echo off
:begin
echo abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHI
goto begin:
```

Endnotes

[1] P-States are defined by the Advanced Configuration and Power Interface. See Section 2.6 of revision 3.0b of the specification at http://acpi.info/DOWNLOADS/ACPIspec30b.pdf . More information on P-states specific to Intel processors is available in Chapter 13 of *Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3A: System Programming Guide, Part 1* at http://download.intel.com/design/processor/manuals/253668.pdf

[2] Intel Dynamic Acceleration is described in section 13.3 of *Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3A: System Programming Guide, Part 1* at http://download.intel.com/design/processor/manuals/253668.pdf and also in section 2.4.2 of the datasheet *Intel® Core™2 Duo Mobile Processor, Intel® Core™2 Solo Mobile Processor and Intel® Core™2 Extreme Mobile Processor on 45-nm Process* at http://download.intel.com/design/mobile/datashts/32012001.pdf

[3] Dynamic FSB Frequency Switching is described in section 2.4.1 of the datasheet *Intel® Core™2 Duo Mobile Processor, Intel® Core™2 Solo Mobile Processor and Intel® Core™2 Extreme Mobile Processor on 45-nm Process at* http://download.intel.com/design/mobile/datashts/32012001.pdf

[4] Speedstep technology is described in Section 13.1 of *Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3A: System Programming Guide, Part 1* at http://download.intel.com/design/processor/manuals/253668.pdf and also in section 2.2 of the datasheet *Intel® Core™2 Duo Mobile Processor, Intel® Core™2 Solo Mobile Processor and Intel® Core™2 Extreme Mobile Processor on 45-nm Process* at http://download.intel.com/design/mobile/datashts/32012001.pdf

[5] Software-controlled Clock Modulation is described in section 13.5.3 of *Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3A: System Programming Guide, Part 1* at http://download.intel.com/design/processor/manuals/253668.pdf

[6] See the specification for T_J (Junction Temperature) in Table 22 on p. 103 of the datasheet *Intel® Core™2 Duo Mobile Processor, Intel® Core™2 Solo Mobile Processor and Intel® Core™2 Extreme Mobile Processor on 45-nm Process at* http://download.intel.com/design/mobile/datashts/32012001.pdf

[7] See section 5.1.2 of the datasheet *Intel® Core™2 Duo Mobile Processor, Intel® Core™2 Solo Mobile Processor and Intel® Core™2 Extreme Mobile Processor on 45-nm Process* at http://download.intel.com/design/mobile/datashts/32012001.pdf

[8] The ACPI standard in effect at the time Dell released the Latitude E6500 was revision 3.0b, available online at http://acpi.info/DOWNLOADS/ACPIspec30b.pdf

[9] See Chapters 8 and 11 of the revision 3.0b ACPI specification at http://acpi.info/DOWNLOADS/ACPIspec30b.pdf

[10] See *Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3A: System Programming Guide, Part 1* at http://download.intel.com/design/processor/manuals/253668.pdf and *Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3B: System Programming Guide, Part2* at http://download.intel.com/design/processor/manuals/253669.pdf

[11] See section 13.5.3 of *Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 3A: System Programming Guide, Part 1* at http://download.intel.com/design/processor/manuals/253668.pdf

[12] See p. 31 at http://support.dell.com/support/edocs/systems/late6500/en/sqrg/pdf/U084C0MR.pdf

[13] See p. 6 at http://support.dell.com/support/edocs/systems/late6400/en/sfit/Nu0312D.pdf