COMPAQ

White Paper

February 2002
167T-0202A-WWEN
Prepared by:
Workstations Division
Compaq Computer Corporation

Contents

Hyper-Threading Technology, New Feature of Intel Xeon Processor

Abstract: *New refreshes of Compaq Evo Workstations W6000 and W8000 will feature the leading edge Intel Xeon processor with Hyper Threading technology. This new processor is fabricated with the latest .13µ* (micron) *technology, 512KB-L2 cache, and the ability to support frequencies ranging from 1.8 GHz to greater than 2.6 GHz. This processor also includes support for multi-threaded execution, known as Hyper-Threading technology.*

The purpose of this paper is to provide an overview of Hyper-Threading technology, to analyze and study the performance impacts of the system with Hyper-Threading technology, enabled as well as disabled, and to provide the technical information and benefits of this feature.

Notice

Hyper-Threading Technology, New Feature of Intel Xeon Processor
White Paper prepared by Workstations Division

First Edition (February 2002)
Document Number 167T-0202A-WWEN

Introduction

The *Compaq Evo* Workstations W6000 and W8000 incorporate the new leading-edge Intel Xeon Multi-processor (MP) processor and the Intel i860 chipset to achieve the next level of system performance. The purpose of this paper is to explain the impacts on processor performance with the new Intel Hyper-Threading (HT) technology *enabled* versus *disabled* in *Compaq Evo* Workstations W6000 and W8000.

The performance of the system is directly related to Instructions Per Clock (IPC). The two ways to increase system performance today are the following.

- The operating system (OS) of multi-processor (MP) system configurations, such as Microsoft Windows NT, Windows 2000, Windows XP and any later OS that supports MP, take advantage of Thread Level Parallelism (TLP). These operating systems divide their workloads into processes and threads, which can be independently scheduled and dispatched to run on the processors that are idle or available.

Note: Microsoft Windows 98 and Microsoft ME do not support multiple processors.

- Instruction Level Parallelism (ILP) increases the IPC. Very Long Instruction Word (VLIW) and Explicitly Parallel Instruction-set Computing (EPIC) are designed from the ground up to support IPC. However, these instructions do not maintain binary compatibility with the Intel Architecture (IA), known as the IA-32 Instruction-set.

Hyper-Threading Technology

Intel has developed a new feature called Hyper-Threading technology for the IA-32 architecture, which provides a performance boost to the processors.

Hyper-Threading technology signals a fundamental shift in processor design. The multi-threading design techniques allow an OS to view a single physical processor as if it were two logical processors. To accomplish this, processors enabled with Hyper-Threading technology can manage incoming tasks from different software applications and continuously alternate from one task to the other, without losing track of the data processing status of each task.

Current processor technologies queue up individual instructions and handle them one at a time in a logical order (out of order execution, in-order retirement), much like a person who is trying to watch television while talking on the phone. In the current scenario, the person takes in data from each medium, handling it piece by piece. Hyper-Threading technology allows the viewer to accomplish both activities simultaneously without impacting the quality or speed of either experience.

Advantages

Some of the advantages of Hyper-Threading technology include the following:

- Improved overall system performance
- Increased number of users that a platform can support (server applications)
- Improved reaction and response time because tasks can be run on separate threads
- Increased number of transactions that can be executed

For workstation applications that use a lot of memory intensive, multi-tasked or resource bound tasks, there seems to be no apparent benefit to using Hyper-Threading technology. This will be evident from the results of some benchmarks in this paper. There seems to be more benefit running two physical processors as opposed to running a single processor with Hyper-Threading enabled.

Overview

Hyper-Threading technology enables a single physical processor to appear as two independent Logical Processors to the OS. This enables the OS to execute two separate code streams (called threads) concurrently, either from two different applications or from the same application. After power up and initialization, each logical processor can be individually halted, interrupted or directed to execute a specified thread, independently from the other logical processor on the chip.

Unlike a traditional dual processor (DP) configuration (see Figure 1) that uses two separate physical IA-32 processors (such as two Intel Xenon processors), the *logical processors* (see Figure 2) in a processor with Hyper-Threading technology share the execution resources of the processor core, which include the rapid execution engine, the caches, the system bus interface, and the firmware. Each *logical processor* has its own set of general purpose registers (including a separate Program Counter and local Advanced Programmable Interrupt Controller [APIC]) but, in order to minimize the complexity of the technology, the Intel Hyper-Threading technology does not attempt to simultaneously fetch/decode instructions corresponding to two threads. Instead, the Central Processing Unit (CPU) will alternate the fetch/decode stages between the two logical CPUs and only attempt to execute operations from two threads simultaneously, thus addressing the problem of poor execution unit utilization.

Hyper-Threading is available in a Simultaneous Multi-Threaded (SMT) class processor, which has *dual Architectural State*[1]. Simply stated, there are two logical processors on one die. Therefore, two threads can be launched simultaneously on the same processor, which reduces overhead on the thread-switches. The *Architectural State*, which includes the associated register set for the second logical processor, is only about 5% of the total die area.

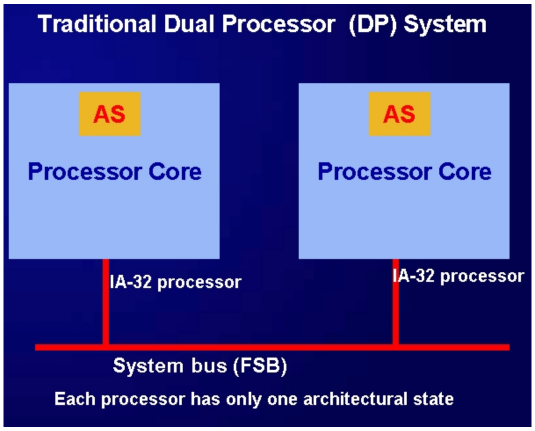


Figure 1

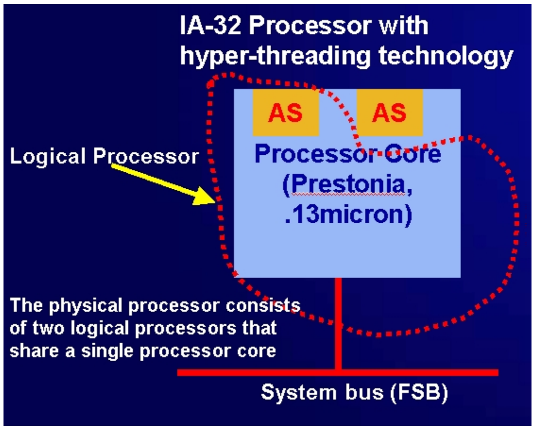


Figure 2

[1] Architectural State represents the current thread context that consists of the IA-32 registers that are visible to the programmer such as data registers, segment registers, control registers, debug registers, and most of the MSRs as well as its own APIC. The conventional microprocessor such as P3 provide only one set of AS. These single threaded processors are used to support multiple threads application today. However, before another thread can begin, the current thread's state must be saved in the memory so it can properly resume later. Depending on the number of registers involved and cache misses incurred, a thread-switch operation involving saving and restoring registers can take hundreds of cycles. Consequentially, it is unprofitable to support thread switching on the operations that take less than a hundred or so cycles.

Using Hyper-Threading technology, Instruction-Level Parallelism can boost performance from a processor by utilizing otherwise idle resources. (See Figure 3.) For simplicity, assume that there are only three Executing units: A, B, and C, and only two threads: T1 and T2. In a single processor system, usually the T2 thread has to wait until the T1 thread is finished since the system can only execute one thread at a time. The MP system has more than one processor that the OS can use to launch TLP, so the T1 thread runs on one CPU and dispatches the T2 thread to another CPU.

Hyper-Threading technology can allow up to two active threads in the pipeline. Since it shares the same rapid execution engine, each thread will need to arbitrate to use the resources in case of conflict. The T1 thread with μop (micro-op) A1 will need to compete with the T2 thread because both threads would have to use the same execution unit A at the same time. Hyper-Threading will give the same priority to each thread and depending on how the code is written, (using Spin-loops [8]*) one will win over the other. Ideally, it could perform as well as MP system configuration but, on an average, Intel claims that there is a 30% performance boost in some applications.

Note: *Numbers in brackets throughout this paper refer to documents listed in the *References* section of this paper.

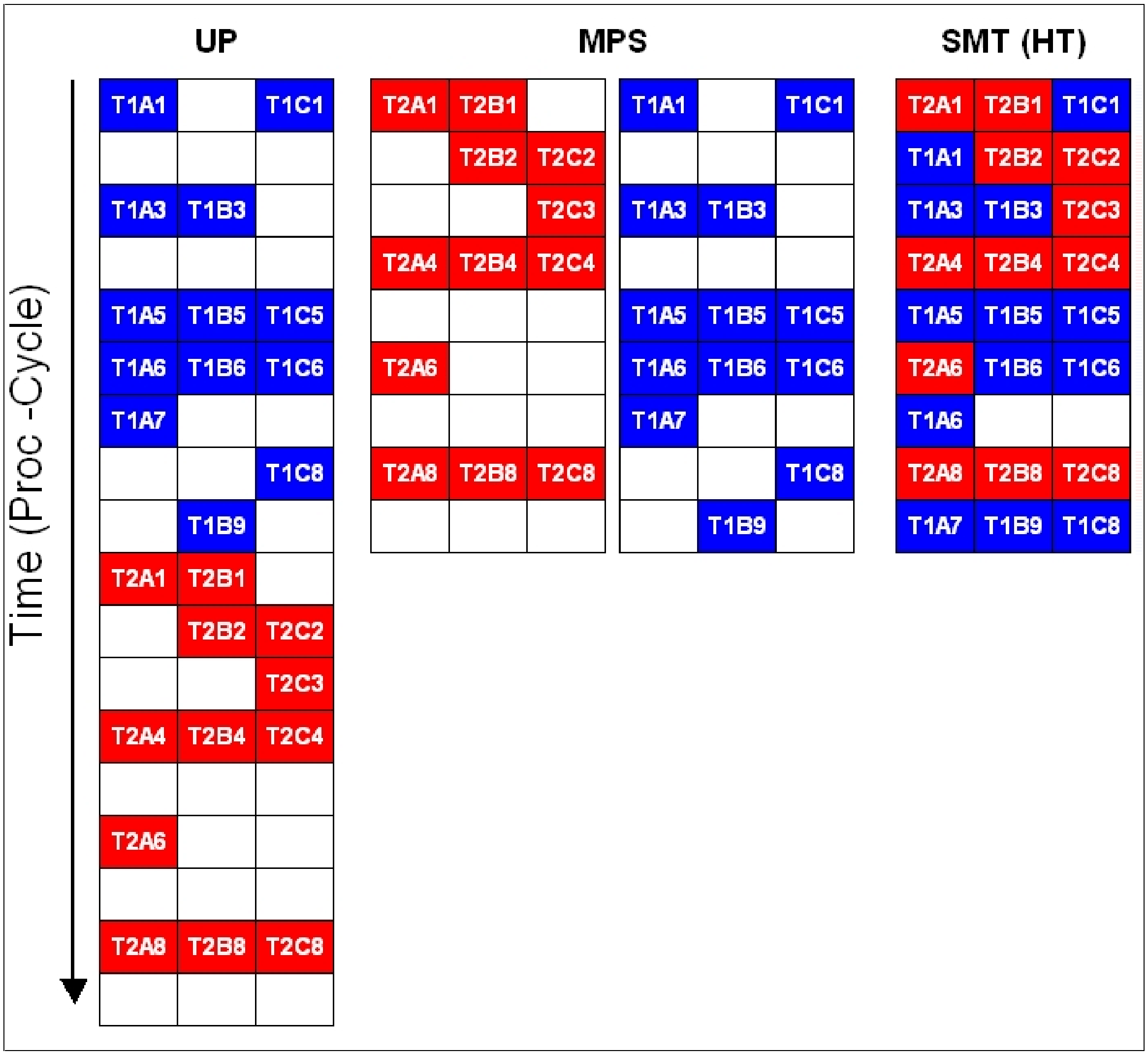


Figure 3

Figure 4 shows the micro-architecture block diagram of the Intel Pentium 4 processor. A thread is initialized by the OS in the same way as in a MP system. There is no distinction in the OS between a logical processor and a true physical processor. The two active threads are interleaved at the instruction fetch and issue stages of the pipeline. (These are the first six blocks on the left in Figure 4.) The trace cache (decoded instruction cache) has an additional tag bit for each instruction, to represent the thread number (0 or 1). If one thread takes an exception, the instructions from only that thread are flushed. The rapid execution engine (the Scheduler, Floating–Point (FP-RF), Integer–RF, and arithmetic logic unit [ALU] stages) is competitively shared between two threads with equal priority, which can be an issue when a low-priority task is scheduled simultaneously with a high-priority, time-critical task. The rapid execution engine is the key to the performance gain. Higher utilization of these resources improves the IPC. Instructions from both threads are simultaneously dispatched for execution by the processor core. The processor core executes these two threads concurrently, using out-of-order instruction scheduling to keep as many of its execution units as busy as possible during each clock cycle. Reorder/retire stage again alternates between two logical processors to commit state in program order.

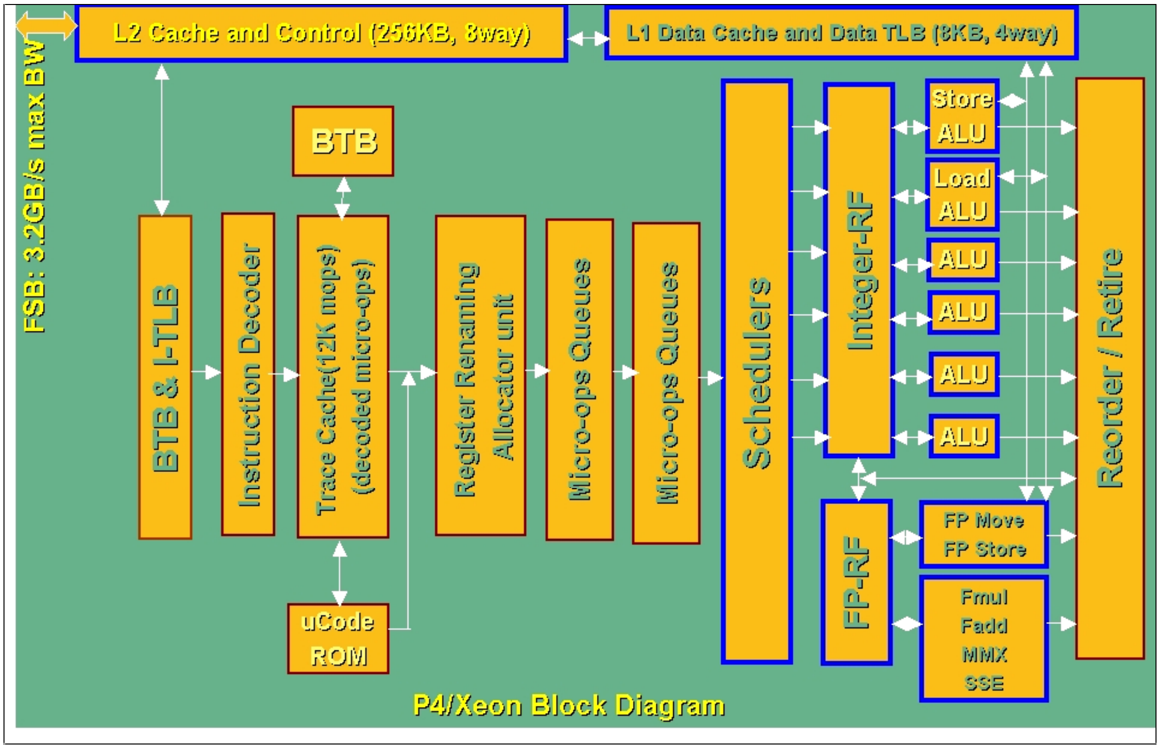


Figure 4:

In a Hyper-Threaded processor, the processor resources are shared so that each thread uses a different resource during each clock cycle. The goal is that the additional thread will fill the voids in the pipeline and take over processor resources when one thread is stalled, while waiting on a cache miss to be filled. With this limitation, some applications will not benefit from the Hyper-Threading technology.

A majority of single-threaded applications will not benefit. This can be seen from the Sysmark and Winstone Benchmark results. (See the *Performance Analyses* section for details.)

Additionally, some multi-threaded applications that are not specifically optimized for Hyper-Threading technology might also show negative or no benefit. For example, the interleaved thread will contend for resources and destroy the effectiveness of the tight code optimizations. Both threads may also come into conflict over precious space in the trace cache.

Applications such as Internet audio and video streaming, high-performance 3D graphics, digital photography and digital video encoding/decoding, speech recognition, multimedia, MP3 encoding, and database applications do benefit from Hyper-Threading technology. Naturally, these applications are already multi-threaded and each thread is handling a different task that works well with a Hyper-Threaded processor. In addition, these applications can take advantage of the new 142 SSE2 new instructions. These new instructions reduce the overall number of instructions required to execute a particular program task and, as a result, can contribute to an increase in overall system performance.

Compaq Evo Workstations W6000 and W8000

The *Compaq Evo* Workstations W6000 and W8000 incorporate the new leading-edge Intel Xeon MP processors and the Intel i860 chipset to achieve the next level of system performance. A detailed discussion of the Xeon processor and the i860 chipset, including performance advantages, can be found in the *Compaq* white paper [6] 1527-0501A-WWEN (April 2001) *Architectural Comparison of Compaq Evo Workstation W6000/W8000 with Compaq Workstation AP550/SP750*.

Compaq Evo Workstations systems arrive with Hyper-Threading technology disabled. This is done for several reasons.

First, in order to see the real advantage of Hyper-Threading technology, most applications need to be rewritten and recompiled for TLP and ILP. Recompiling with Streaming Single Instruction Multiple Data [SIMD] Extensions (SSE2) can deliver additional performance gains. Not many of the applications that are available today are recompiled to take advantage of HT. Therefore, the majority of the existing applications will perform better with the two physical CPUs versus two logical CPUs, which Hyper Thread Technology provides.

Secondly, the system with Microsoft Windows 2000 Professional Version and dual physical processors cannot take advantage of Hyper-Threading technology since the Professional version only supports up to two CPUs. Only the server version of Windows 2000 supports Hyper Threading technology.

Finally, Windows NT does not support Hyper-Threading technology.

IMPORTANT: If a system was originally shipped with one CPU, the system may have an Advanced Configuration and Power Interface *(ACPI) Uniprocessor* Hardware Abstraction Layer (HAL). You will need to switch to *'ACPI Multiprocessor'* HAL in order to use Hyper-Threading technology.

Enabling/Disabling Hyper-Threading

This section outlines the steps to enable and disable the Hyper-Threading technology on *Compaq Evo* Workstations W6000 and W8000.

Preliminary Steps

Before enabling or disabling the Hyper-Threading technology, perform these preliminary steps:

1. Check the firmware version at the boot up screen in the lower left corner to ensure that version *v01.12* is installed.

Note: If the firmware version is earlier than v01.12, then it is possible that the CPU in the system does not have support for Hyper-Threading technology.

---

2. Flash the system with the firmware version *v01.15* or the latest available on the Web.

IMPORTANT: If you are not familiar with the *Compaq* ROM-based setup, refer to *Appendix A* for detailed procedures.

---

3. Check to see if the *ACPI Multi-processor PC* is installed on the system. See *Appendix B* for details to check and switch to a different HAL.

After completing the preliminary steps, follow these steps to enable or disable the Hyper-Threading technology on the *Compaq* Workstation W6000 and Workstation W8000:

1. Reboot the system.
2. When the screen appears with the red *COMPAQ* logo and the F10 prompt at the bottom of the screen appears, press **F10** to enter the setup mode. Once the system completes Power On Self Test (POST), your system will go into setup mode.
3. When prompted for the language selection, select the appropriate language and press **Enter**.
4. Move to the Advanced Tab. Select **Device Options** and **Enter**.
5. Scroll to the Hyper-Threading option and use the left or right arrow key to toggle between enable/disable.
6. Once the desired option is selected, press **F10** to save that option and exit that window.
7. Press **F10**, press **Enter**, press **F10** to save changes, and **Exit.**

The system saves your configuration and reboots.

Hyper-Threading and Microsoft Windows XP

If the Hyper-Threading technology option is enabled, verify that the change was made. With Hyper-Threading enabled under Microsoft Windows XP, any physical CPU will be seen by the OS as two logical CPUs.

1. Verify by opening the Task Manager.
2. Right-click on the task bar, and select **Task Manager** to bring up the Windows Task Manager.
3. Under the Performance tab, you will see four CPUs in the Task Manager if your system has two Physical CPUs and Hyper-Threading is enabled.

Figure 5 shows the Task Manager of a system that has two physical Processors with Hyper-Threading enabled. Figure 6 shows the Task Manager of a system that has either one physical processor with Hyper-Threading enabled or a system that has two physical processors with Hyper-Threading disabled.

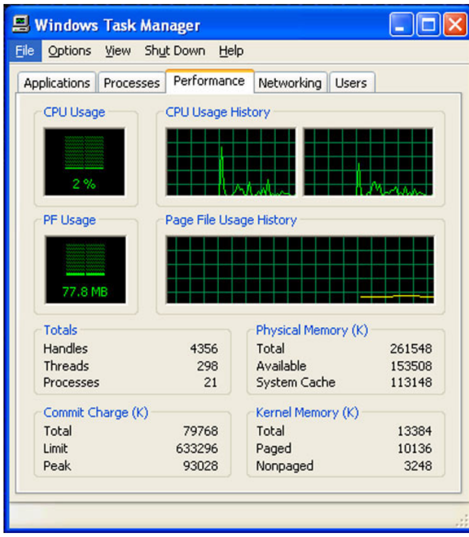


Figure 5:

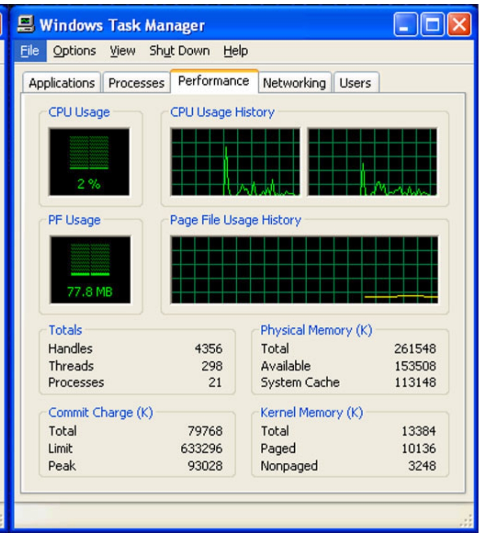


Figure 6:

Figure 7 shows the Task Manager on a system that has only one processor with Hyper-Threading disabled.

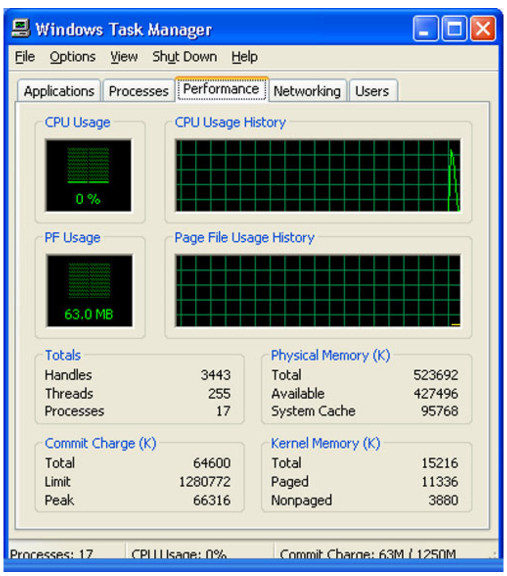


Figure 7:

Hyper-Threading and Microsoft Windows 2000

Microsoft Windows 2000 Professional Edition only supports up to a maximum of two processors. The OS can only recognize two CPUs, even-though the system has two physical CPUs that support Hyper-Threading and have Hyper-Threading enabled. The OS does not recognize the case of one physical CPU with Hyper-threading enabled versus the case of two physical CPUs, regardless of Hyper-Threading enabled or disabled. The Task Manager in Figure 8 reflects these three cases. The first case with two logical CPUs will have less performance than the other two cases. Figure 9 shows a single processor system with Hyper-Threading disabled.

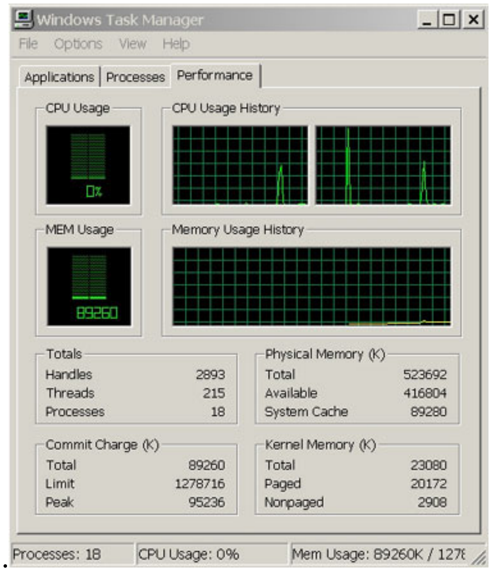


Figure 8

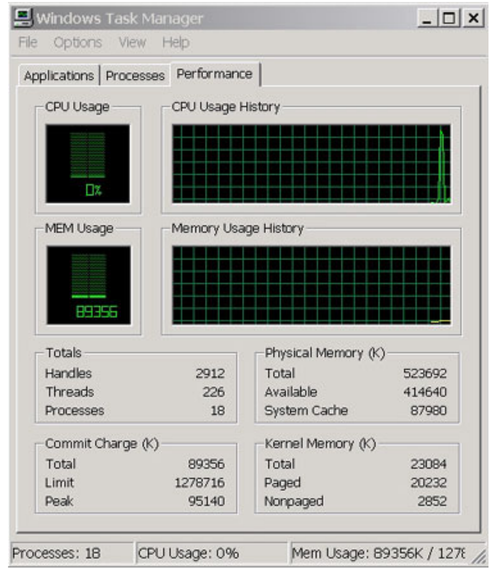


Figure 9:

Performance Analyses

Applications generally need to be tuned and recompiled[2] to gain the maximum performance with Hyper-Threading technology. Most of the benchmarks that are used to collect the performance data in this paper are horizontal and single threaded. The purpose of this paper is to explain the impacts of processor performance with Hyper-Threading technology enabled versus disabled.

SYSmark 2001

Productivity performance should represent current business usage models of multi-tasking with background computing. The industry standard productivity benchmark SYSmark 2001 incorporates mainstream applications for office productivity and Internet content creation as well as the latest business usage models to reflect platform productivity performance.

SYSmark 2001 is a suite of application software and associated benchmark workloads developed by the Business Applications Performance Corporation (BAPCO), a non-profit consortium of leading computer industry publications, independent testing labs, PC hardware manufacturers, semiconductor manufacturers, and software publishers. SYSmark 2001 is a tool that measures system performance on popular business-oriented applications in using the Microsoft Windows operating environment.

SYSmark 2001 contains fourteen application workloads that are divided into the following two categories:

Office Productivity:

- Dragon Naturally Speaking Preferred Version 5
- McAfee Virus Scan 5.13
- Microsoft Access 2000
- Microsoft Excel 2000

[2] Intel is working to provide Hyper-Threading development tool for software optimization and multi-threaded code that tune for NetBurst microarchitecture processors with Hyper-Threading Technology

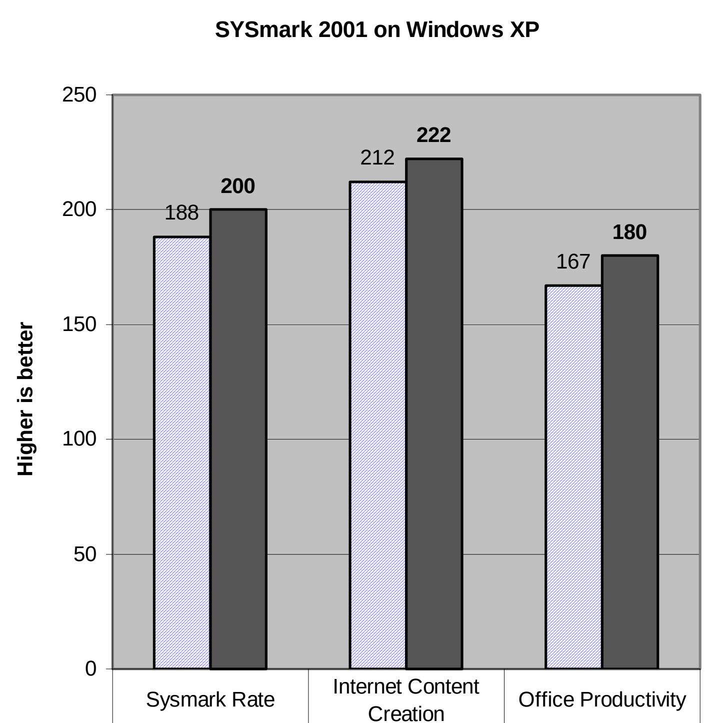
- Microsoft Outlook 2000
- Microsoft PowerPoint 2000
- Microsoft Word 2000
- Netscape Communicator 6.0
- WinZip 8.0

Internet Content Creation:

- Adobe Photoshop 6.0
- Adobe Premiere 6.0
- Macromedia Dreamweaver 4
- Macromedia Flash 5
- Microsoft Windows Media Encoder 7

Analysis

SYSmark 2001 is a horizontal benchmark and runs only one application at a time. Therefore only the application thread is running for any given window time. To run the same benchmark on the true MP system does not yield any performance gain or degradation since most of the time the other processor is idle. In this case, when the Hyper-Threading technology is enabled to emulate multi-processing, performance is degraded about 5% since the application thread now is shared and competing with the OS threads for the same processor resources. That does not happen with a true multiprocessor system since each of the processors has its own resources. See Figure 10.



	Sysmark Rate	Internet Content Creation	Office Productivity
Enable	188	212	167
Disable	200	222	180

Figure 10

Winstone

Business Winstone 2001

Business Winstone 2001 is a system-level, application-based benchmark that measures the overall performance of a PC when running today's top-selling Windows-based 32-bit applications on Microsoft Windows 98 SE, Windows NT 4.0 (SP6 or later), Windows 2000, Windows Me or Windows XP. Business Winstone 2001 does not mimic what these packages do. It runs real applications through a series of scripted activities and measures the time a PC takes to complete those activities to produce its performance scores. Higher scores mean better performance. When Business Winstone 2001 runs the test, it runs at least a portion of each application through a script that eTesting Labs developed. The script automatically executes commands within that application with no input from the user.

Business Winstone 2001 uses the following applications in its tests:

- Norton Antivirus 2000 from Symantec
- WinZip 7.0
- Microsoft FrontPage® 2000
- Lotus Notes R5
- Microsoft Access 2000
- Microsoft Excel 2000
- Microsoft PowerPoint 2000
- Microsoft Project 98
- Microsoft Word 2000
- Netscape Communicator 4.73

Content Creation Winstone 2001

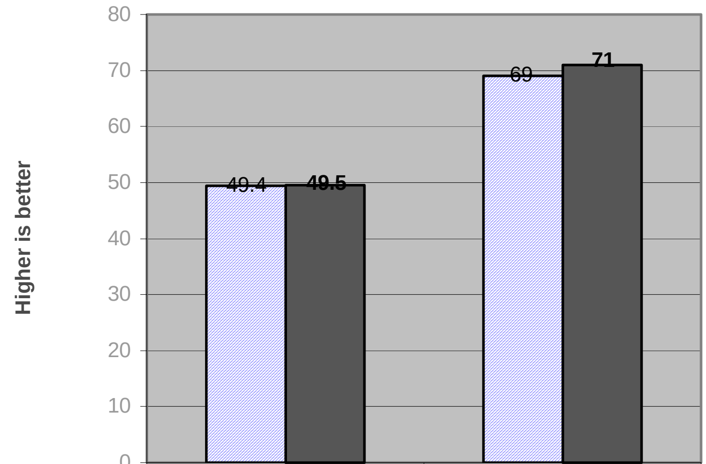
Content Creation Winstone is a system-level, application-based benchmark that measures a PC's overall performance when it is running under a 32-bit OS, such as Windows 2000 or Windows XP. Content Creation Winstone 2001 uses the following applications:

- Adobe Photoshop 5.5
- Adobe Premiere 5.1
- Macromedia Director 8.0
- Macromedia Dreamweaver 3.0
- Netscape Navigator 4.73
- Sonic Foundry Sound Forge 4.5

Following the lead of real users, Content Creation Winstone 2001 keeps multiple applications open at once and switches amongst those applications. Content Creation Winstone 2001 is a single large test that runs the above applications through a series of scripted activities and returns a single score. Those activities focus on so called "hot spots," periods of activity that make your PC really work—the times where you are likely to see an hourglass or a progress bar.

Analysis

The Winstone test results are the same as the results for SYSmark 2001. No performance loss or gain is seen with Hyper-threading enabled. See Figure 11.

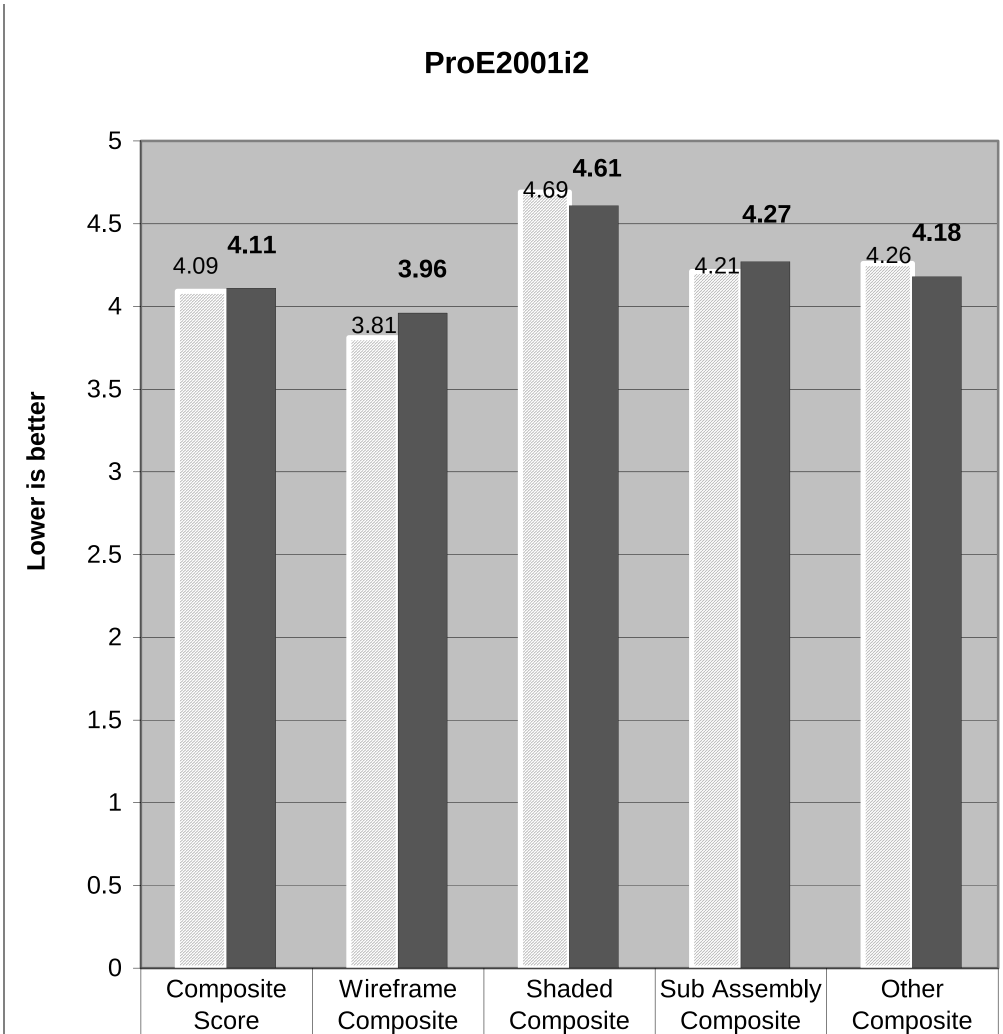


	Business	Content Creation
Enable	49.4	69
Disable	49.5	71

Figure 11

Pro/ENGINEER 2001i²

The Standard Performance Evaluation Corporation (SPEC) SPEC/GPC Application Performance Characterization (SPECapc) project group offers performance results and free downloads for a benchmark based on Pro/ENGINEER 2000i² (ProE 2000i²). However, the results for SPECapc for Pro/E 2000i² cannot be compared to those of previous versions of the benchmark. The model used in the benchmark is a realistic rendering of a complete photocopy machine consisting of approximately 370,000 triangles. See Figure 12.



	Composite Score	Wireframe Composite	Shaded Composite	Sub Assembly Composite	Other Composite
Enable	4.09	3.81	4.69	4.21	4.26
Disable	4.11	3.96	4.61	4.27	4.18

Figure 12

The benchmark is comprised of 17 tests. Startup and initialization time is measured, but given no weight (0.0) within the composite score for the benchmark. There are 16 graphics tests, each of which measures a different rendering mode or feature. The first three graphics tests measure wire-frame performance using the entire model. The next four measure different aspects of shaded performance, using the same model. Each of these tests executes exactly the same sequence of 3D transformations to provide a direct comparison of different rendering modes. The next four tests use a sub-assembly, and compare the two FASTHLR modes, the default shading mode, and shading with edges. These tests also execute a common sequence of 3D transformations. The last five graphics tests use two different instances of the model—the first three without its outer skins (to illustrate the effect of FASTHLR and level-of-detail operations), and the last two to illustrate complex lighting modes and surface curvature display. The last test is an aggregate of all time not accounted for by the previous 16 tests, and is a mix of CPU and graphics operations.

Scores are generated for all 17 tests. Composite numbers are provided for each set of graphics tests (shaded, sub-assembly, wire-frame, and other) and there is an overall composite score for graphics and CPU operations. Start-up and initiation time is not included in the composite score.

You must have a 3D graphics display device recognized by Pro/E 2000i² in order to run the benchmark. A fully licensed, released version of Pro/E 2000i² is required. If a floating license is used, the workstation's network must be configured as documented in the Pro/ installation guide.

Analysis

ProE 2002i2 is not a multi-threaded application but does contain different tasks for comparing. Wireframe and sub-assembly, which are more computationally intensive, take more advantage of Hyper-Threading technology by utilizing the idle processor resources. Otherwise, shading, texturing, and Wireframe, which are more memory intensive, see a negative performance impact due to competitively shared caches. A trace cache miss results in a memory latency that forces the execution unit to use the X86 instruction decoder, instead of the μops (micro-operation) that are already decoded in the trace cache.

Excel Benchmark

Excel benchmark uses the Microsoft Excel application to measure the performance of computer system. Like other benchmarks, the Excel benchmark measures performance by managing a large amount of data. A lower score indicates better system performance. The Excel benchmark includes four tasks. Each task will have roughly about 40MB of data. Excel benchmark will run four tasks at the same time from a batch file. When running from a batch file, there will be four Excel windows open. The first Excel window does the sorting. All columns are sorted in ascending and descending order 15 times. The second Excel window does the multiplication. Two sheets are multiplied together 400 times. The third Excel window does the Cosine of data in a nested loop 24 times and sorts the data in ascending and descending order. The fourth Excel window sorts the amount of data in ascending and descending order 22 times. When all four tasks finish running, the run time of each task is recorded. The longest time is chosen for performance comparisons.

Analysis

Excel benchmark is a multi-task application and each task utilizes different resources from each other. This type of application is ideal for Hyper-Threading technology. Performance gain from a single physical processor to dual physical processors is about 80% since dual physical processors do not have to share and compete for the same cache. The result is that the system has twice as large a cache when compared to the single physical processor with Hyper-Threading enabled. For a single physical processor with Hyper-Threading enabled, there is a 30% performance gain, whereas an 18% performance increase in a dual physical processor case is realized with Hyper-Threading enabled. This is because the application is not as computationally intensive as others for utilizing all of the processor resources. See Figure 13.

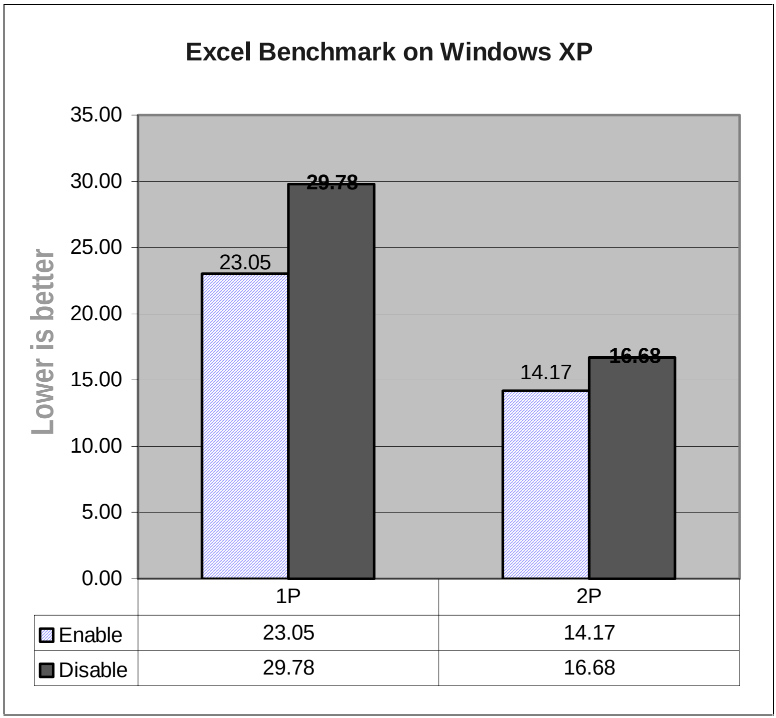


Figure 13:

Solid Edge V10

SPECapc for Solid Edge V10 was developed by EDS PLM Solutions in cooperation with the SPECapc group. It represents typical user operations that are valuable in evaluating the performance of systems running Solid Edge. The new benchmark runs on Microsoft Windows NT and Windows 2000. Two Solid Edge models are used for the benchmark:

- A well-head assembly with 1,320 parts
- A 1.3-million polygons and a 64-part hydraulic jack assembly with 66,000 polygons

Solid Edge V10 measures the following four basic areas of performance:

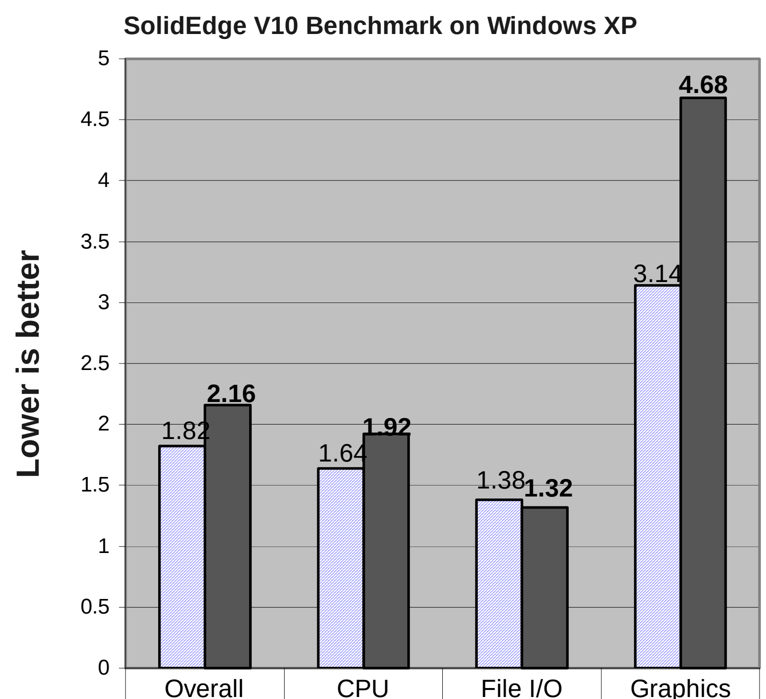
- **Graphics**—measures commonly used commands in three modes:
  - Smooth shading
  - Wireframe
  - Smooth shading with textures and reflections
- **File I/O**—measures opening and saving the well-head assembly model using three different display configurations:
  - All inactive
  - All active—a combination of active, inactive, and hidden parts
  - **CPU**—exercises real-time VHL representation and re-computing an assembly
- **Workflow**—represents a typical set of operations that users will execute during the course of creating a project:
  - Real-time drawing view placement

- o In-place assembly activation
- o In-place part activation
- o Modification of a model
- o Update of all the files due to the model change

Note: EDS PLM Solutions consider all four portions of the benchmark to be equally important in evaluating performance using Solid Edge, so they have been given equal weightings when calculating the geometric mean.

Analysis

From this benchmark, the results show that the SolidEdge V10 CPU and graphics tests are computation intensive and will benefit from the Hyper-Threading technology. The file I/O test again has to share and compete for resources with the alternate threads, including the OS threads that require file I/O operations and thus performs poorly. See Figure 14.



	Overall	CPU	File I/O	Graphics
Enable	1.82	1.64	1.38	3.14
Disable	2.16	1.92	1.32	4.68

Figure 14:

Montecarlo Benchmark

The Montecarlo utility includes four independent tasks that run at the same time. Each task contains four spreadsheets. Each spreadsheet contains different Montecarlo functions, formula, and graphic calculations.

Analysis

The Montecarlo application is multi-tasked as well as computationally intensive. A task could utilize all of the processor resources, which is determined by the fact that the ratio between the physical 1P (30.1) and physical 2P (15.15) is almost 2:1. Independent of the 1P or 2P case, Hyper-Threading technology does not yield any performance gain since the benchmark is resource bound. See Figure 15.

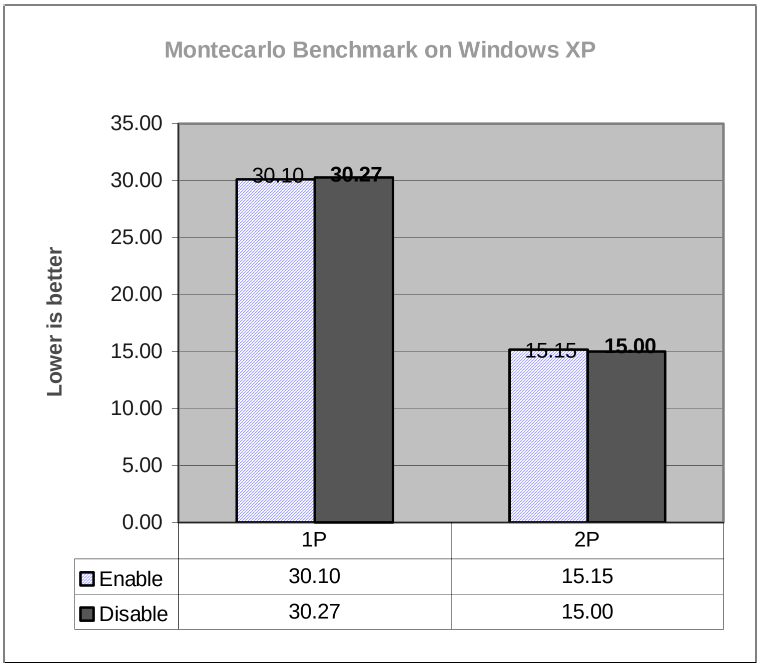


Figure 15:

Verilog Compiled Simulation Benchmark

The Verilog Compiled Simulation (VCS) benchmark script includes four independent tasks that run at the same time. Each task simulates the same reset sequence of a memory controller. The VCS benchmark is very memory intensive.

Analysis

This VCS application is multi-tasked as well as memory intensive. A task requires about 20MB of data for simulation. The data cache is shared between the two threads, which results in frequent cache line replacement and hence shows poor performance when Hyper-Threading is enabled. In a two physical processor case with Hyper threading disabled, with cache available on each processor, there is a significant improvement in performance. With Hyper-Threading technology enabled in a dual processor configuration, the performance is worse than just a true dual processor configuration. This poor result is due to the sharing of resources between the same processor and checking the other processor's cache validity for correct data. See Figure 16.

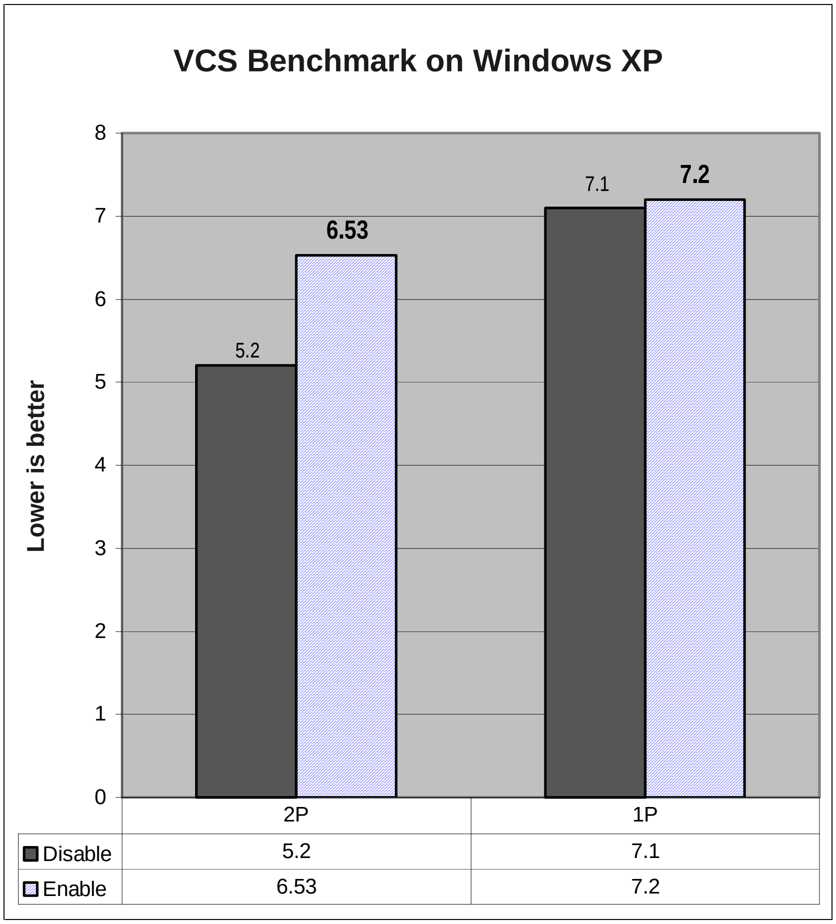


Figure 16

Summary

Hyper-Threading technology represents a new approach to improving the instruction throughput of processors (instruction-level parallelism) that is targeted for servers and high-performance workstations. One of the principal benefits of Hyper-Threading technology is that it enables the processor to use on-die resources that may otherwise be idle. Although existing operating system and application code will run correctly on a processor with Hyper-Threading technology, to obtain performance gain, some relatively simple code modifications are needed to get the optimum benefit from Hyper-Threading technology. To refrain from wasting resources with the idle loops, the OS needs to issue immediately the Halt instruction to the logical processor that has a thread going idle. A thread that is using spin lock code, which aggressively captures processor resources, may need to add the pause instruction within the loop to allow the other threads a fair access to the processor resources. The limited trace cache size requires tight code loops in the trace cache for Hyper-Threading technology to be effective. Programmers now avoid loop unrolling, as it increases the amount of code created and increases trace cache misses. A trace cache miss forces the execution to the X86 instruction decoder, which can only decode one X86 instruction per cycle.

On the down side, the existing single threaded applications do not experience the benefit of Hyper-Threading technology. In fact, there can sometimes be disruption by the alternative thread, such as the thread that uses a spin lock without pause instruction. Multi-threaded applications that are optimized for MP systems also need to consider resource contention that destroys the effectiveness of hand-tuned highly optimized code (games and streaming media applications).

Hyper-Threading technology can provide a performance gain of up to 30% when executing under multi-threaded OS and applications with code that is tuned for Hyper-Threaded3 technology over a comparable IA-32 processor without Hyper-Threading technology. For workstation applications that are more multi-tasked and more memory intensive, there is more benefit to using two physical processors as compared to using a single Hyper-Threading technology enabled processor.

[3] Tuned instructions for Hyper-threaded is minimize the resource contention

Appendix A — Enable/Disable Hyper-Threading

IMPORTANT: Check and verify that your system was flashed with the latest *Compaq ROMPAQ* version v01.15 or later, as shown in the lower left corner at the boot screen. Download the latest *ROMPAQ* at the *Compaq* website http://www.compaq.com

---

To enable or disable the Hyper-Threading technology on the *Compaq Evo* Workstations W6000 and W8000, follow these steps:

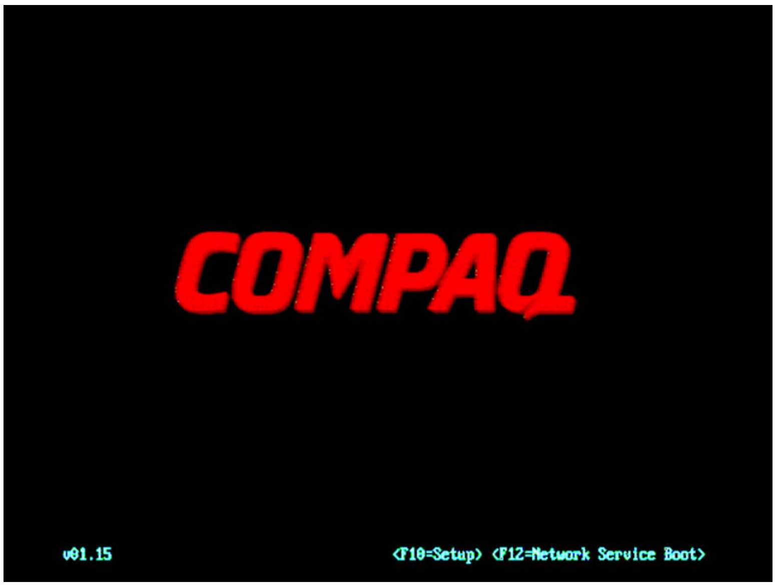


Figure 17:

1. When you power-up the system, you see the screen shown in Figure 17. Press **F10** to enter the setup option. After a few seconds, your screen goes into the setup mode.
2. Select the language according to your region by using the **Up** or **Down** arrow key. In this case, select **English** and press the **Enter** key. See Figure 18.

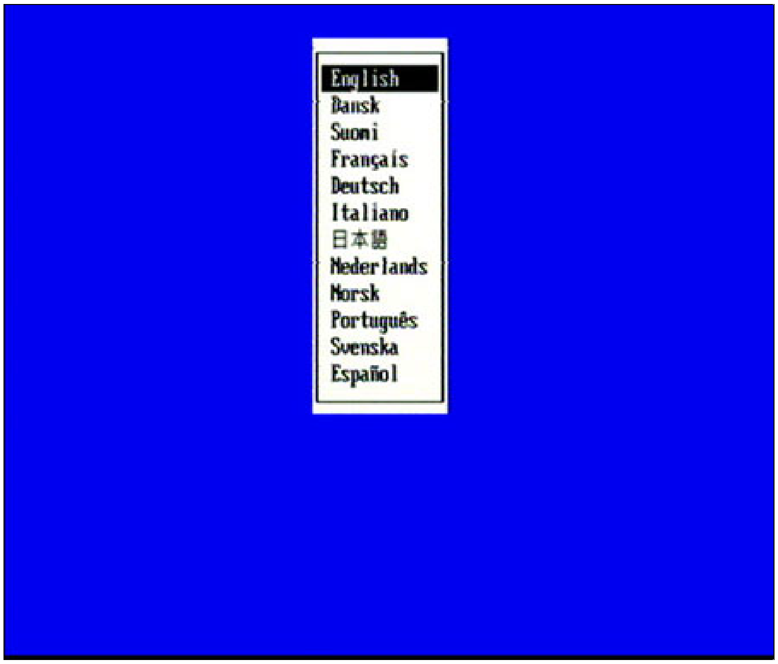


Figure 18:

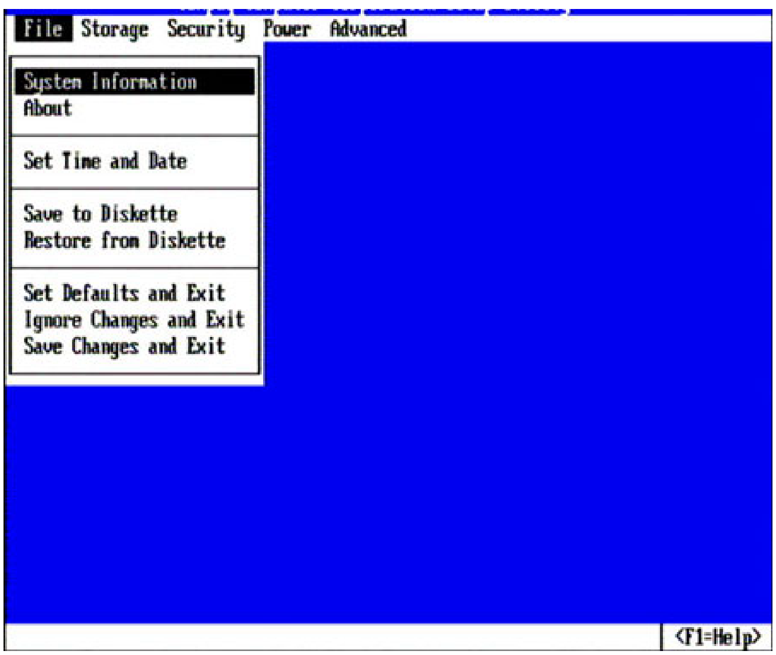


Figure 19:

3. Use the **Right** arrow key to select the **Advanced** tab. See Figure 20.

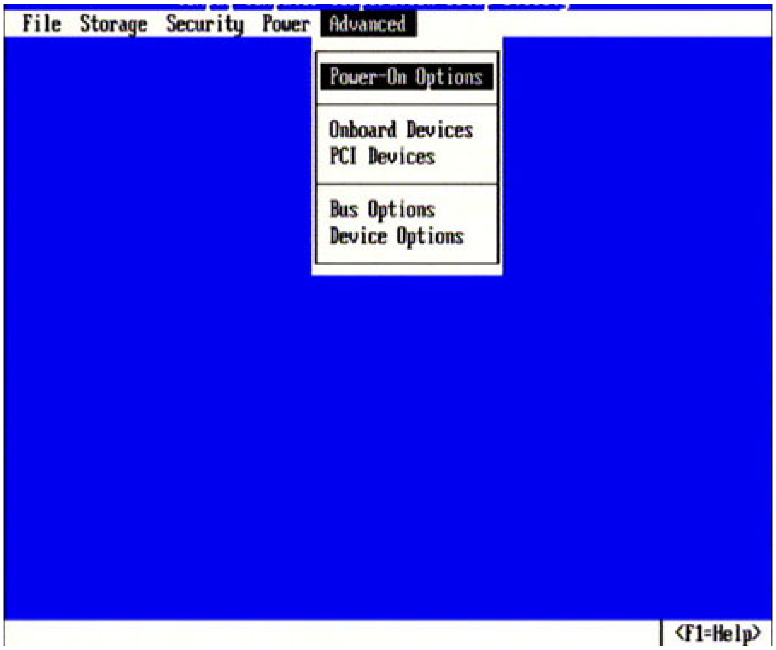


Figure 20:

4. Use either the **Up** or **Down** arrow key to select **Device Options**. See Figure 20.
5. Press the **Enter** key to open the **Device Options** setting window. See Figure 21.

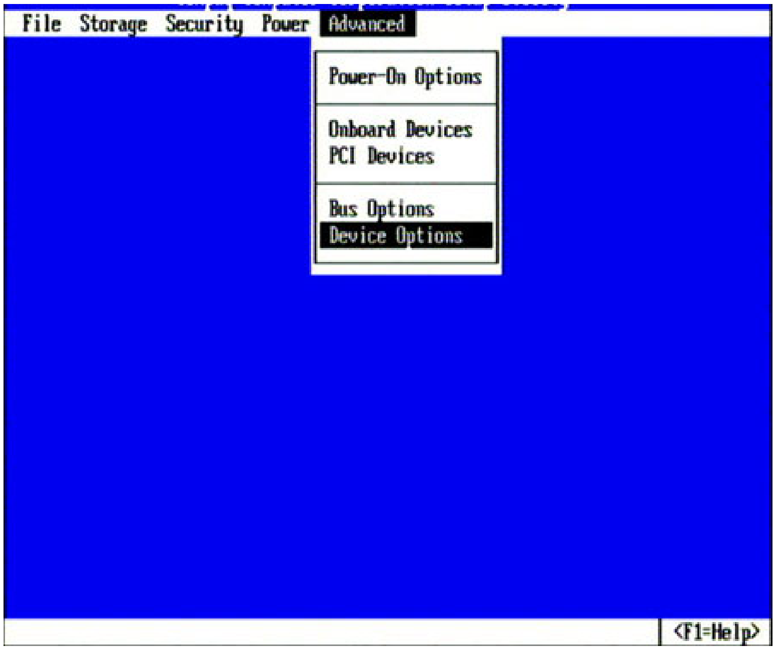


Figure 21:

6. Use the **Up** or **Down** arrow key to move the white arrow cursor to select **Hyper-Threading**. See Figure 22.

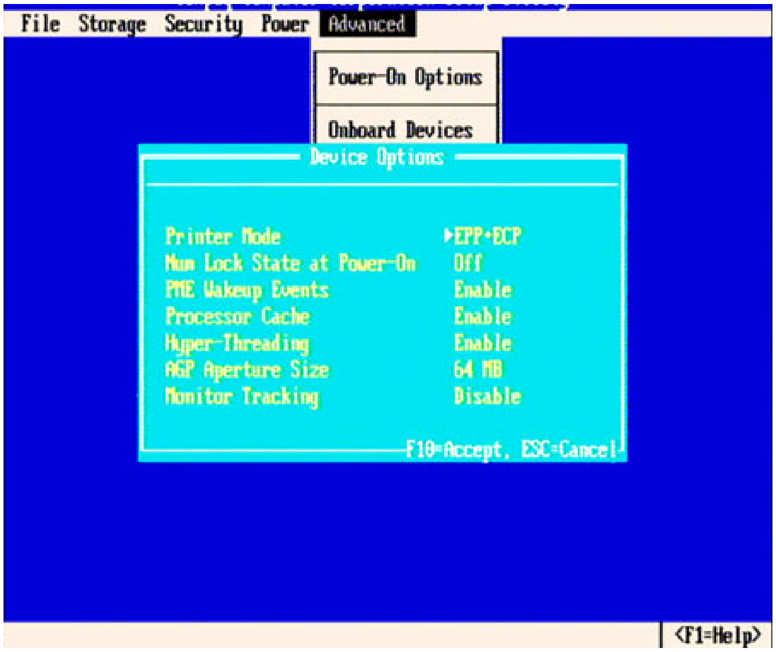


Figure 22:

7. Use the **Left** or **Right** arrow key to toggle between **Enable** or **Disable** Hyper-Threading, depending on the option that you choose. See Figures 23 and 24.

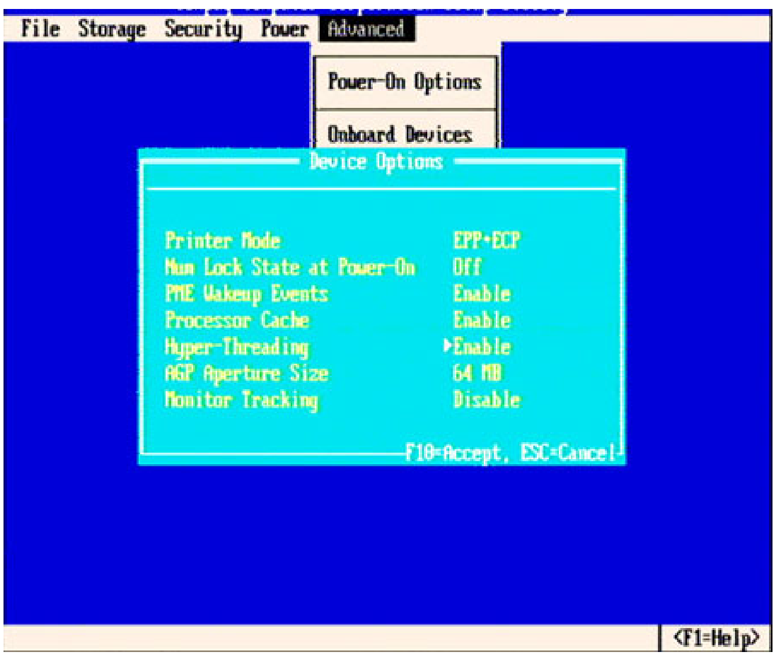


Figure 23:

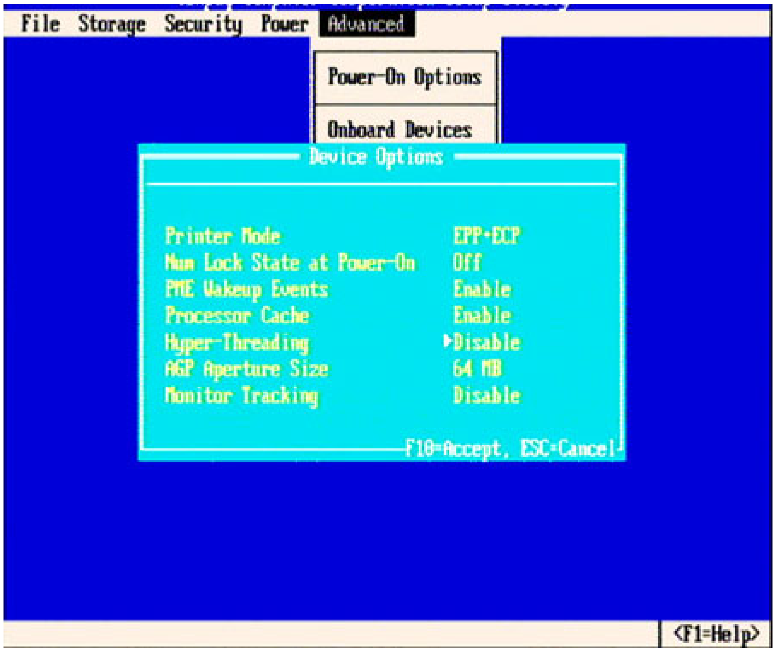


Figure 24:

8. Press **F10** to close the Device Options window.
9. Press **F10** to select **Save Changes** and **Exit**. See Figure 25.

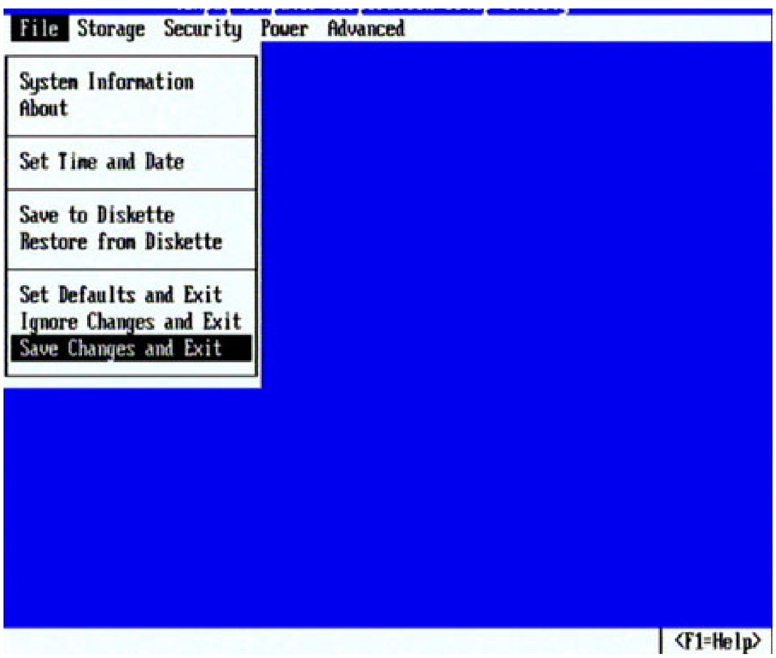


Figure 25:

10. Press **Enter** to save any changes. Press **F10** again. See Figure 26.

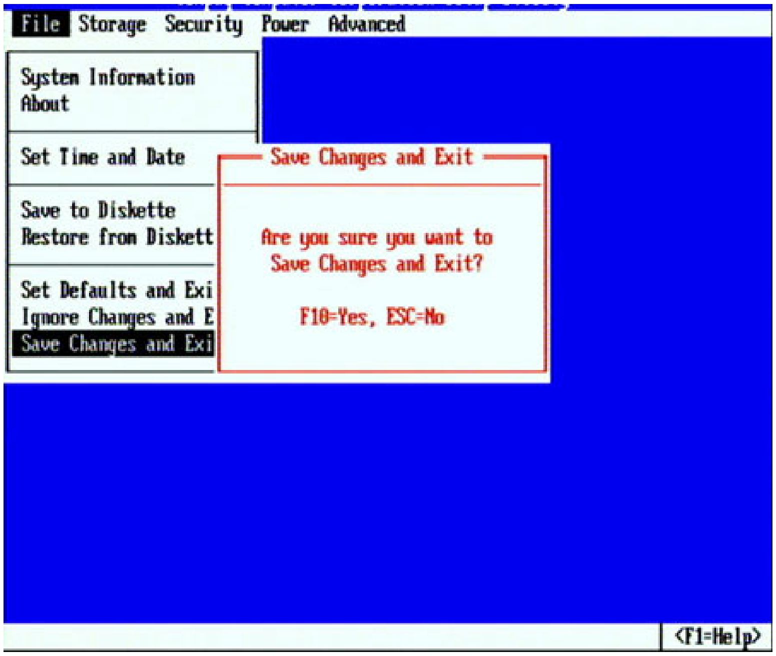


Figure 26:

11. Your system saves the configuration and is ready for reboot. See Figure 27.

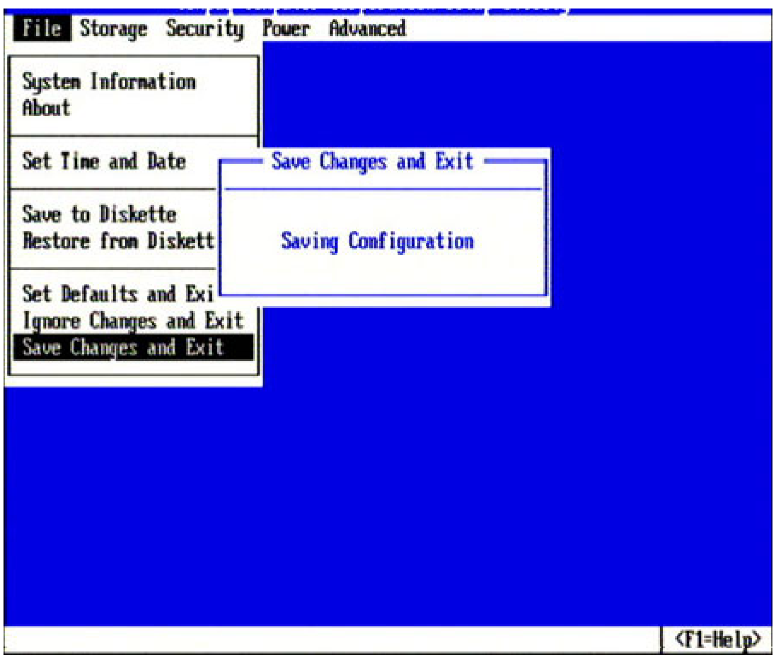


Figure 27:

Appendix B — Verify HAL Version

This section will describe how to verify which version of the HAL has been installed on your system and how to switch to *'ACPI Multiprocessor PC'* HAL. The procedure is for Windows 2000.

Note: The procedure for Windows XP is not included here since it is similar to the Windows 2000 procedure.

Follow these steps to verify which version of the HAL has been installed on your system:

1. At the Desktop screen, right click on **My Computer** and move the mouse to the **Properties** bar. See Figure 28.

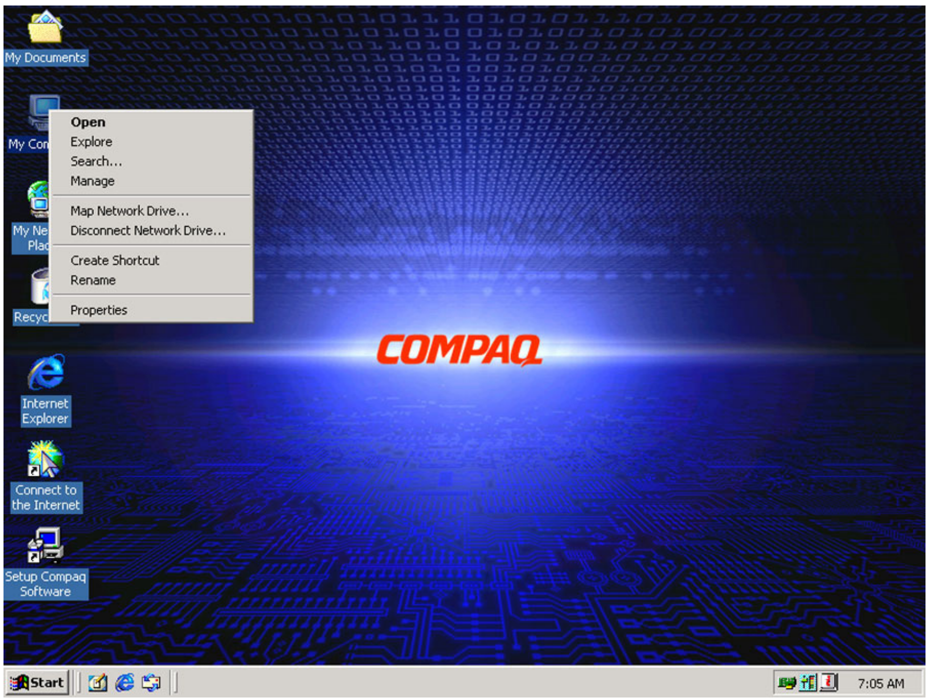


Figure 28:

1. **Click** with the left mouse button to bring up the **System Properties** window. See Figure 29.

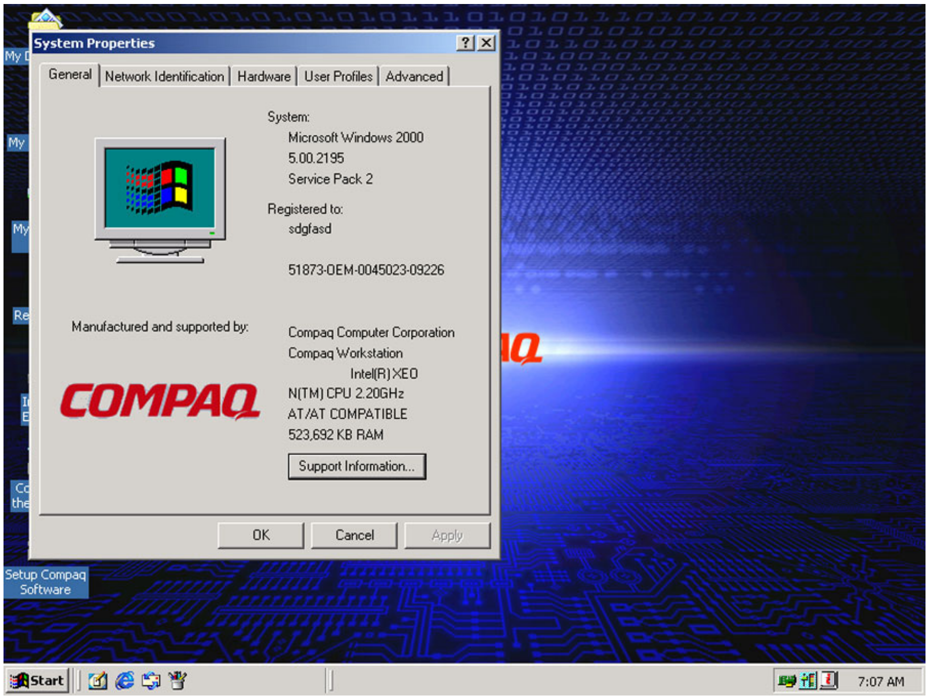


Figure 29:

3. Click on the **Hardware** tab and **Device Manager** button to bring up the Device Manager window. See Figure 30.

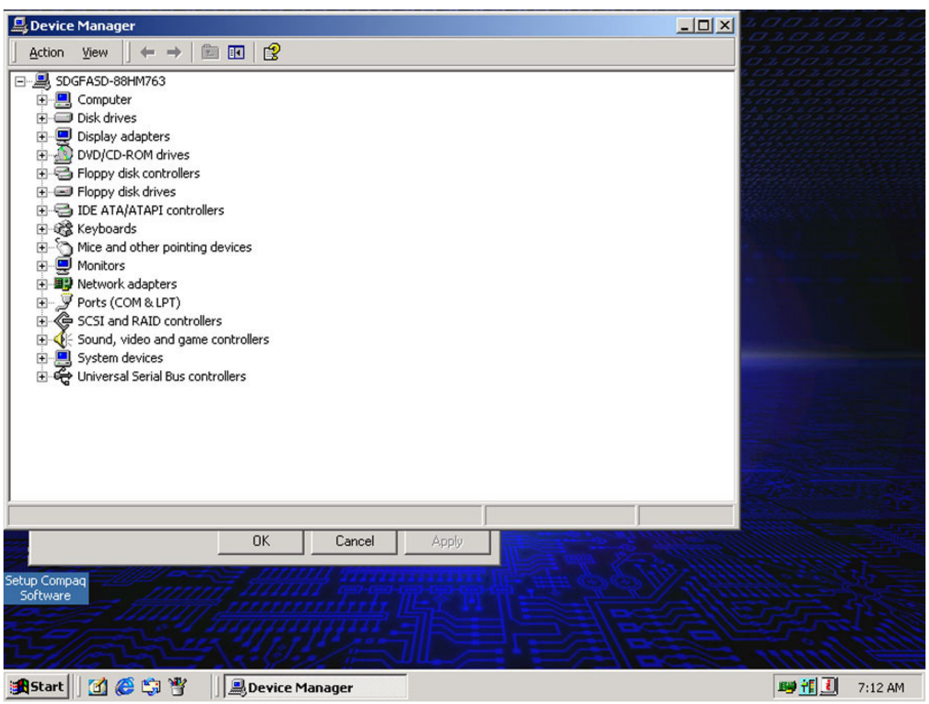


Figure 30:

4. Double click on the **Computer Icon**. If you see "*ACPI Multiprocessor PC*" is installed, then the system is ready.

Note: You could stop by closing all open windows and returning to the Desktop.

If you see "*ACPI Uniprocessor PC,*" then follow this procedure:

1. Right click on ***ACPI Uniprocessor PC*** and **click** on the **Properties** bar to bring up the *ACPI Uniprocessor PC Properties* window. See Figure 31.

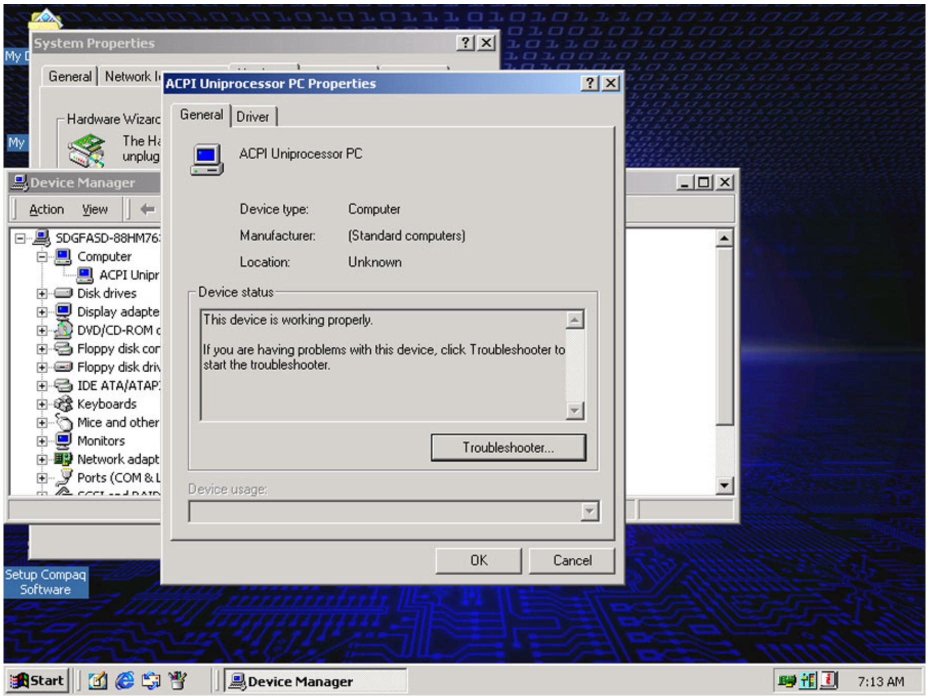


Figure 31:

2. Click on the **Driver** tab and select **Update Driver** button to bring up the Upgrade Device Driver Wizard window. See Figure 32.

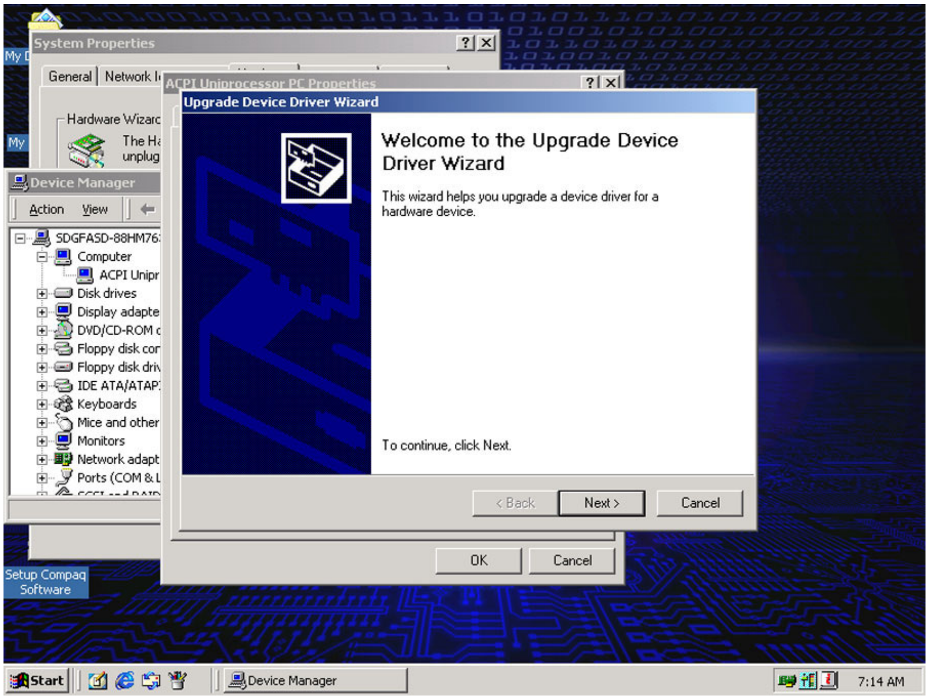


Figure 32:

3. Click **Next**. See Figure 33.

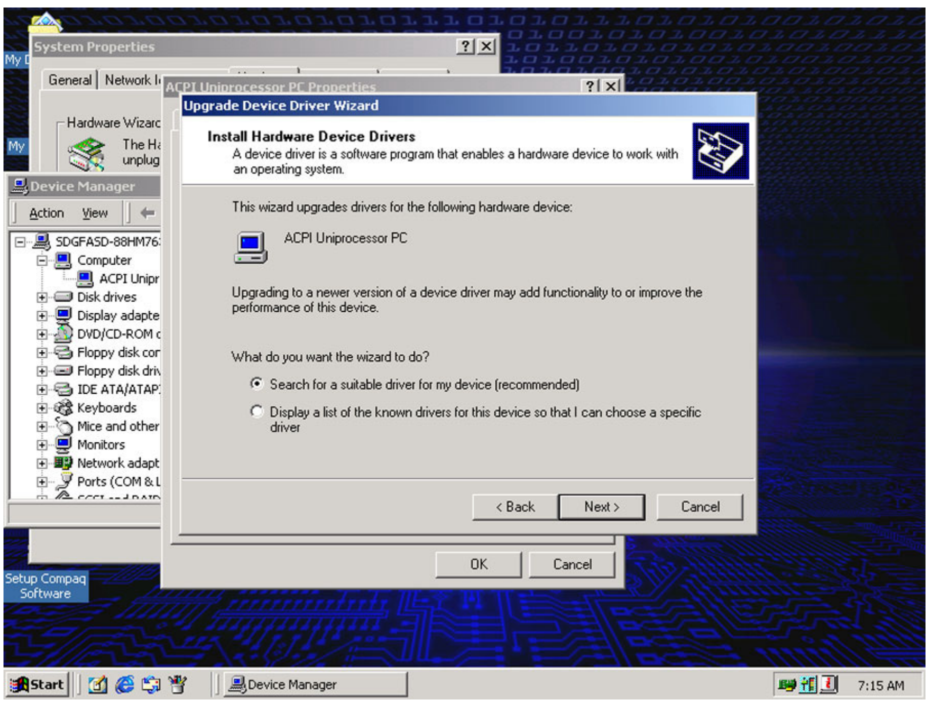


Figure 33:

4. Select the second option to **Select a Device Driver** and click **Next**. See Figure 34.

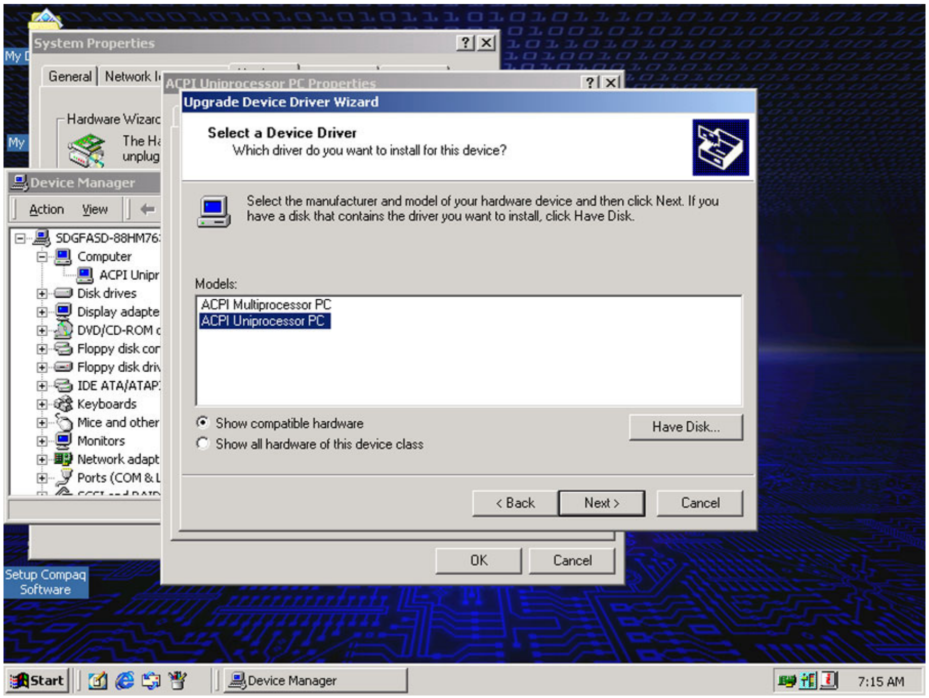


Figure 34:

5. Highlight **ACPI Multiprocessor PC** and click **Next**. See Figure 35.

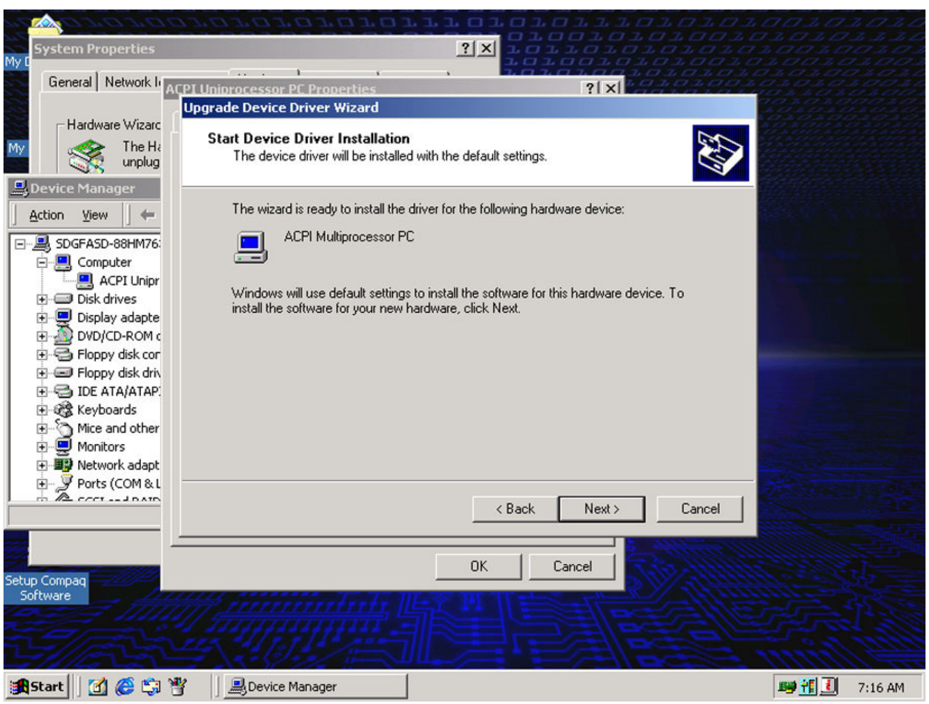


Figure 35:

3. The Upgrade Device Driver Wizard window prompts you that the new device is ready to install; in this case, it is the *ACPI Multiprocessor PC* kernel. Click the **Next** button to continue.

4. Click **Finish** to close the Upgrade Device Driver Wizard window. See Figure 36.

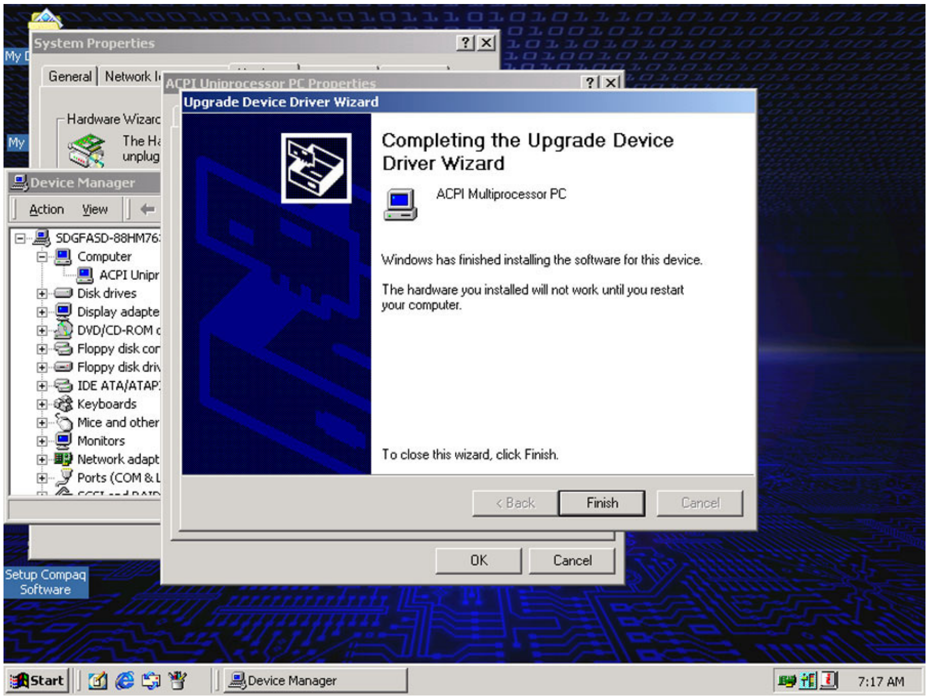


Figure 36:

5. To close the *ACPI Multiprocessor PC Properties* window, click **Close**. See Figure 37.

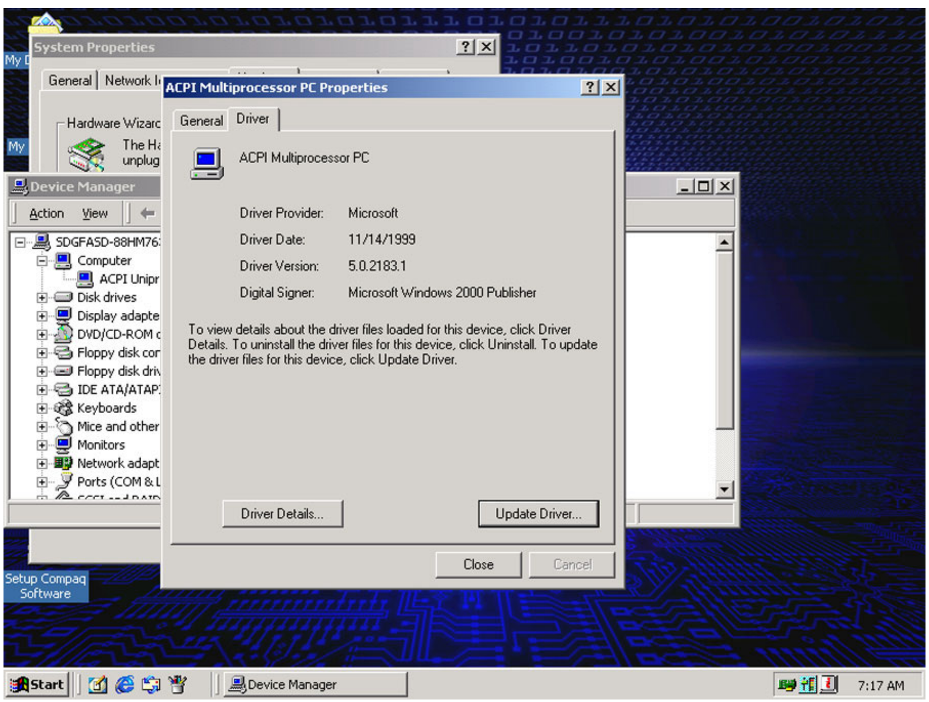


Figure 37:

6. At this time, the system is ready for rebooting. Click **Yes** to restart the system. See Figure 38.

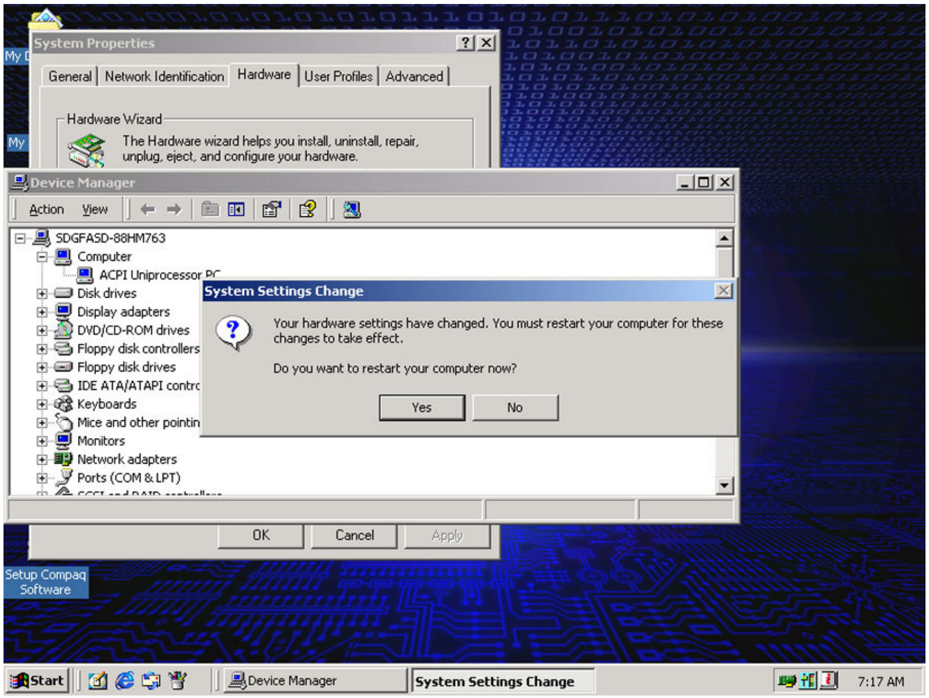

System Properties
General
Network Identification
Hardware
User Profiles
Advanced
Hardware Wizard
The Hardware wizard helps you install, uninstall, repair, unplug, eject, and configure your hardware.
Device Manager
Action
View
SDGFASD-88HM763
Computer
Disk drives
Display adapters
DVD/CD-ROM drives
Floppy disk controllers
Floppy disk drives
Keyboards
Monitors
Network adapters
Ports (COM & LPT)
System Settings Change
Your hardware settings have changed. You must restart your computer for these changes to take effect.
Do you want to restart your computer now?
Yes
No
OK
Cancel
Apply
Setup Compaq Software
Start
Device Manager
System Settings Change
7:17 AM

Figure 38:

References

[1] Introduction to Hyper-Threading Technology, Intel Document Number: 250008-002

[2] Laudon J., Gupta A., Horowitz M. (1994) Interleaving: A Multi-threading Technique Targeting Multi-processors and Workstations

[3] Smith B. J. (1985) The Architecture of HEP

[4] Thistle M., Smith B.J. (1988) A Processor Architecture for Horizon

[5] Alverson R., Callahan D., Cummings D., Koblenz B., Porterfield A., Smith J.N. (1990) The Tera Computer System

[6] -0501A-WWEN (April 2001) Architectural Comparison of Compaq Evo Workstation W6000/W8000 with Compaq Workstation AP550/SP750

[7] Tullsen D.M., Eggers S.J., Levy H.M. (1995) Simultaneous Multi-threading: Maximizing On Chip Parallelism

[8] AP-949 Using Spin-loops on Intel Pentium 4 Processor and Xeon Processor