DOCKETED

Docket Number:	14-AAER-02
Project Title:	Computer, Computer Monitors, and Electronic Displays
TN #:	204806
Document Title:	Intel and Dell Comments on Aggiosâ€™ Report
Description:	N/A
Filer:	Harinder Singh
Organization:	California Energy Commission
Submitter Role:	Commission Staff
Submission Date:	6/1/2015 1:05:57 PM
Docketed Date:	6/1/2015

CEC Draft Computer Standards (Docket #12-AAER-2)

Intel and Dell Comments on Aggios' Report[1]

In Appendix C Notebook Demonstration Project
Phase 3 - Notebook Power Management Analysis and Demonstration
Part I - Opening up the Power Management (Version 1.0; submitted 10/27/2014)

Key Contacts:

Shahid Sheikh: Intel
Gary Verdun: Dell

Dated: May 29, 2015

---

[1] The original Appendix C contained references which were lost during conversion from PDF to Word document

Overview

California Energy Commission (CEC), in its Draft Staff Report for Computers (dated March 12, 2015) has heavily relied on a technical report submitted by Aggios Inc. on October 27, 2014 as an Appendix C, in an addendum[1] to an earlier submittal dated August 6, 2013. CEC developed its cost effectiveness model for computers, based on cost/benefit analysis found in this report. Much of the work in Appendix C is speculative and lacks understanding of ground realities. Intel and Dell collaborated to provide an initial detailed technical response in the following pages (section by section). We expect other Industry members will provide their comments on the report later.

Industry welcomes the opportunity to further educate CEC and other stakeholders to clear misperceptions resulting from Aggios' report. CEC's heavy reliance on this report for its staff draft proposal is a serious concern to us, and we are committed to work with CEC to set the record straight.

---

[1] http://www.energy.ca.gov/appliances/2014-AAER-01/prerulemaking/documents/comments_12-AAER-2A/California_IOUs_Standards_Proposal_Addendum_Computers_2014-10-27_TN-73899.pdf

Executive Summary

This report provides a qualitative analysis of the energy consumption of desktop and notebook computers in the long and short idle modes based on a similar analysis conducted for IPTV set top box devices.

The goal is to suggest hardware, software and design methodology improvements for the reduction of computer energy consumption by comparing computers and mobile devices in the equivalent and similar idle modes. The proposed solutions are guided by the principle of energy proportionality which requires that the work done by the software and the energy consumed by the hardware are measured and kept in proportions. We are advocating for improvements in the way software is reporting the work it is conducting, the introduction of energy-driven hardware hierarchies, more openness for device integrators to control the power management of the hardware and an industry standard-driven methodology to align software and hardware. Our estimate is that these efforts have the potential to reduce the energy consumption of computers by around 25% in short idle and 50% in long idle. The resulting total savings for the US electricity ratepayers are estimated at $510M per year at the total incremental costs of $54M per year. In order to provide a detailed quantitative analysis of the proposed improvements and define the next steps to move toward the solution, we recommend the initiation of a new project developing a fully functional computer reference design implementing the suggested improvements and conducting detailed measurements, testing and an in-depth costs/benefits analysis. We estimate that such a project would take between 12 and 18 months to complete.

<Dell Comment>:

Author never establishes any relationship between set top boxes and PC systems from a power management perspective. No evidence is ever provided to show that any of the findings in set top boxes are in any way applicable to PC systems.

<Intel Comments>:

There is a fundamental difference between the set top boxes and the general purpose PC's with respect to power management. Set Top Boxes are single purpose device, as such, they do *one* thing. It's fairly trivial for software/hardware/firmware to implement power management functions when it can assume exactly what each and every other piece of HW/SW/FW in the system are doing.

Now compare that with a PC. The biggest hurdle in considering power management solutions in a computer is the fact that it's a general purpose device, an open platform with complex levels of hardware, firmware and software coming from a multitude of different vendors. Its usage model is a variable, along with the loaded software, various add-ons (expansions boards, integrated peripherals, even plug in devices like keyboards/cameras, etc.). It is very difficult for a single "entity" to be able to implement a robust power management solution in this type of environment. For example: it's relatively easy for BIOS, a custom ASIC, an onboard microcontroller, or even a piece of software to setup timers to measure idle times on a given piece of hardware. Once the device is deemed idle for a length of time, the controlling device can put the idle piece of hardware into a low power mode. While this may sound like the ideal solution for closed devices, only the usage model of the specific machine can determine whether it's okay to put an "idle" device to sleep. Any "power management controller" must be aware of each and every custom usage model in order to make intelligent decisions on when and how to put a device to sleep. This is the primary reason why power management was moved away from APM to ACPI. BIOS (APM) can easily measure idle times and put devices to sleep, but only the OS (ACPI) can make an intelligent decision on whether it's a good idea to put a device to sleep. Our key concern with this proposal is that data that is being used is based on a closed, single function device (IPTV set top box device). It is orders of magnitude easier to implement power management solutions on a closed, single-function device, than it is to implement a similar solution on an open, fully-configurable device such as a consumer PC. These numbers are created by extrapolating from a set-top box (single purpose compute device). You simply can't assume you'll get equivalent numbers on general purpose PC's. Based on this, we believe the analysis and conclusions of this paper are simply unfounded.

Introduction

The report describes the forward-looking analysis and ideas for the improvement of the energy consumption of personal computers (desktops and notebooks). It is similar in objectives and methods to the IPTV set top box project conducted previously under the CEC EISG grant, but it is based on qualitative assessments and not on implemented and tested improvements on an actual device. The first two phases of the notebook demonstration project have led the research team to discover that the analyzed notebooks are built without much openness and adaptability in the areas of power and energy management. Despite a variety of approaches, the attempts to control the power states from the application, OS or firmware software side stayed unsuccessful. This report documents the steps taken and discusses the need and justification for a more open and easier to implement power management in personal computers, like desktops or notebooks.

<Intel Comments>:

Intel has been at the forefront of power management of mobile PCs since the beginning (early 1990's). While it is true at that time that ACPI became a foundational change in how PCs are power managed in an open hardware and software environment (moving away from BIOS-based PM in 1995), it is simply untrue that this framework, augmented highly by hardware-driven fine-grained power management, and extensive ecosystem enabling of third party devices/drivers is fundamentally-flawed in anyway. Intel in conjunction with its OEM partners has enabled ~ 20X reduction in platform idle power in systems using Power Optimizer, shipping on 4th generation Core (codenamed Haswell) in 2013, after delivering year over year battery life improvement prior to this. Power Optimizer involved significant augmentation of numerous standards in order to provide the infrastructure necessary for the hardware to take very aggressive and deep power management under the ACPI framework that provides the necessary OS-level info to know when processors and devices are idle. Some examples of the broad and deep efforts to radically optimize the platform for high energy efficiency include:

1. Display Power – Embedded Display Port (eDP) augmented with Panel Self Refresh allowing an integrated LCD panel to refresh itself when the image has not been altered frame-to-frame. This allows for a dramatic reduction in SoC, memory and DP I/O power by allowing the display subsystem to be idle when the display image is non-changing.
2. PCIe – Latency Tolerance Report PCIe peripherals to communicate to the SoC what their latency tolerance is with respect to main memory, allowing the SoC to enter ultra-deep low power states.

Zero power link states such as L1.OFF were added to eliminate the ~10mW link state power associated with a device that is present/idle.

3. SATA – Low power link states as well as zero power link states were added to the standards including DEVSLP# that allows for FLASH storage drives to reach sub 10mW idle power levels.
4. USB2 – LPM L1 was a new state Intel proposed, and was accepted for HS/LS/FS peripherals such that the device may enter a low power state between packets, when nominally unable to enter the USB suspend state. Additionally, numerous features were added to the XHCI (USB controller) to allow for fine-grained deep idle states under conditions where historically platform power management would be inhibited.
5. USB3 – Devices are required to enter low power link states (e.g. U1, U2) when idle, and latency per traffic class is specified by default, or device may inject alternative latencies as require for quality of service (i.e. LTM).
6. RT D3 – Run Time D3 is a new infrastructure for operating systems starting with Windows 8 that allows for motherboard peripherals to be powered off completely when idle, thereby reducing rest of platform (RoP) power.
7. FIVR and high-efficiency platform VRs – With 4th generation Core we enabled on-die fine grained fully-integrated voltage regulation (FIVR) that allows for very fast performance state control (P states) of graphics and IA cores, as well as zero power off states when graphics and IA are idle (RC6, CC6), additional low power package C states as well as chipset deep clock and power gating are supported when latency-tolerance conditions are satisfied (PC6-PC10). In addition to use, the voltage regulator losses are managed by low power signaling from the VR communications channels to allow for voltage regulators (e.g. PMIC, third party discrete, etc.) to know that the system is at a low power state and that significant in-rush current will not be required until the VRs are notified of the impending event and given a pre-defined time for which to recover from. Intel VR efficiency optimizations (e.g. reduce to Vmin, phase-shedding, frequency reduction, etc.).
8. Enabling/tools – Extensive collateral targeted at independent hardware vendors (IHVs) and independent software vendors (ISVs) as well as software test tools that provide deep and broad visibility into a given system designs power management impediments are openly available (VTune and Battery Life Analyzer).

In short, working with the industry Intel achieved the biggest leap forward in battery life and energy efficiency ever in 4th generation Core-based platforms, and continues to innovate in other areas to address power challenges in more active scenarios.

<Dell Comments>:

It should be noted that items in this list with the exception of numbers 7 and 8 are not pervasive in the industry at this time and are predominately focused on Notebook PC implementations. Item 1 for instance is a Notebook only platform implementation as there is no embedded display port in desk top platforms since the only display connections are external to the chassis. It should also be noted that the majority of these implementations required the development of new silicon and often new software to support the additional capabilities. Industry is making these investments in new silicon and software under development and as those devices become available in the market PC systems get designed that include them. It can often take four to six years from the development of a new energy savings technology to the inclusion of that technology in early adopter platforms in the market. Industry continues to invest in energy reductions in all platforms but it is mobile devices where battery life is essential to the end users that drive the investments and associated risks of new reduced energy consumption technology implementations.

These types of enhanced energy reduction technologies are the drivers of lower energy consumption and higher costs for mobile silicon devices relative to standard desk top silicon. It is expected that this differentiation in both cost and energy consumption will continue between desktop and notebook systems in the future.

ACPI and TDP Overview

In January 1992, Intel and Microsoft developed APM (Advanced Power Management) to manage power when a computer system was in idle. At the time, this new technology was driven by the increasing popularity of battery powered notebooks. In December 1996, the successor of APM - the Advanced Configuration and Power Interface (ACPI) specification was developed by Compaq, Microsoft, Intel, Phoenix and Toshiba as the industry open-standard power management interface. The 940 pages long ACPI specification was the consequence of not only the tight cooperation, but also dominance of Microsoft and Intel who controlled the vast majority of personal computers produced. The goal was to remove the third-party BIOS software from the power management activity (and other run-time activities) and establish the Windows OS and its infrastructure as the single system software controlling the run-time of ACPI based computers. The only remaining role of the BIOS software in ACPI based computers became to carry out the boot procedure and then hand over full control to the Windows OS. The side by side comparison of APM and ACPI is shown on Figure 1.

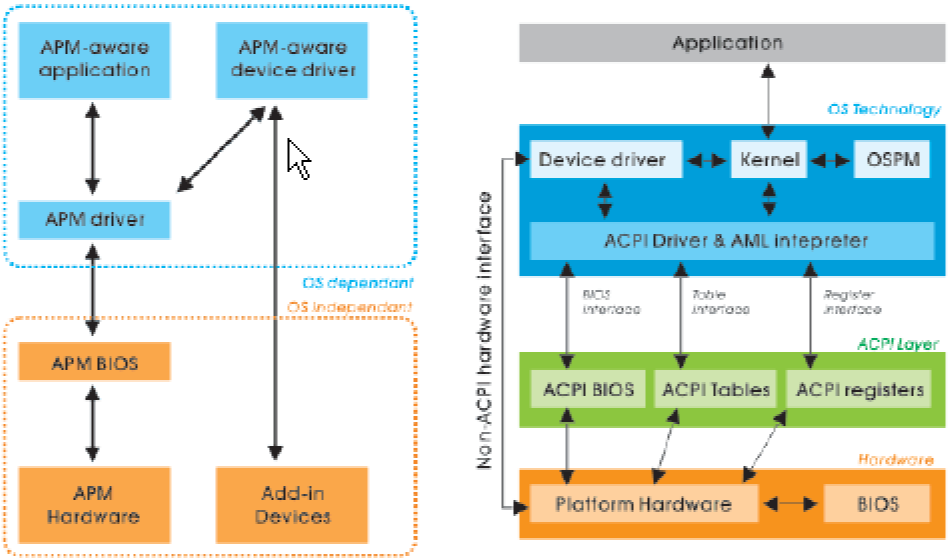


Figure 1 - Comparison of APM (left) and ACPI (right)

The classical BIOS from the APM system was replaced by the OS dependent ACPI layer which was taking over the power management tasks in coordination with the ACPI driver of the OS. As the APM BIOS for each product design has been developed through close collaboration between the BIOS companies and the OEMs, ACPI had to provide a replacement mechanism for the

OEMs to manage the power of the non-standard devices they regularly add to the motherboard and the personal computer, as hard disk drives (HDD).

The ACPI specification states: “*Although the classical fixed hardware programming model requires hardware registers to be defined at specific address locations, the generic hardware programming model used in ACPI allows hardware registers to reside in most address spaces and provides system original equipment manufacturers (OEMs) with a wide degree of flexibility in the implementation of specific functions in hardware. The OS based Power Management (OSPM) directly accesses the fixed hardware registers, but relies on OEM-provided ACPI Machine Language (AML) code to access generic hardware registers. AML code allows the OEM to provide the means for OSPM to control a generic hardware feature’s control and event logic.”* Figure 2 shows the hardware events and control based on ACPI and AML.

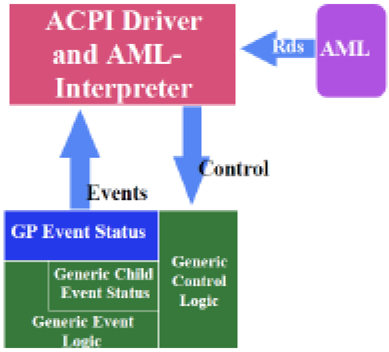


Figure 2 - ACPI and AML for Hardware Control

The specification continues*: “One goal of ACPI is to allow the OEM “value added” hardware to remain basically unchanged in an ACPI configuration. One attribute of value-added hardware is that it is all implemented differently. To enable OSPM to execute properly on different types of value added hardware, ACPI defines higher level “control methods” that it calls to perform an action. The OEM provides AML code, which is associated with control methods, to be executed by OSPM. By providing AML code, generic hardware can take on almost any form.”*

The following is an example provided in the ACPI document: *“As an example of a generic hardware control feature, a platform might be designed such that the IDE HDD’s D3 state has value-added hardware to remove power from the drive. The IDE drive would then have a reference to the AML PowerResource object (which controls the value added power plane) in its namespace, and associated with that object would be control methods that OSPM invokes to control the D3 state of the drive:*

- *_PS0: A control method to sequence the IDE drive to the D0 state.*
- *_PS3: A control method to sequence the IDE drive to the D3 state.*
- *_PSC: A control method that returns the status of the IDE drive (on or off).*

The control methods under this object provide an abstraction layer between OSPM and the hardware. OSPM understands how to control power planes (turn them on or off or to get their status) through its defined Power Resource object, while the hardware has platform-specific AML code (contained in the appropriate control methods) to perform the desired function. In this *example, the platform would describe its hardware to the ACPI OS by writing and placing the AML code to turn the hardware off within the _PS3 control method. This enables the following sequence:*

- *When OSPM decides to place the IDE drive in the D3 state, it calls the IDE driver and tells it to place the drive into the D3 state (at which point the driver saves the device's context).*
- *When the IDE driver returns control, OSPM places the drive in the D3 state.*
- *OSPM finds the object associated with the HDD and then finds within that object any AML code associated with the D3 state.*
- *OSPM executes the appropriate _PS3 control method to control the value-added "generic" hardware to place the HDD into an even lower power state."*

<Intel Comments>:
APM: This is essentially the first generation of power management. APM provided basic power management mechanisms through an interface to BIOS, but it had significant downsides, such as (a) its reliance on BIOS to enter low power CPU states (vs. ACPI where idle state is directly entered by OS) and the BIOS firmware execution (uncached) overhead of such requests, (b) control of only the most basic interfaces (sleep states, hard drive, display, etc.), and (c) the BIOS interface was not compatible with Microsoft Windows® long term OS direction (NT-based vs. Windows® 9x) where the OS was multi-threaded and capable of multi-processing.

ACPI: APM had one big problem alluded to earlier, i.e. BIOS had no sense of what a machine is doing with respect to usage model. For example, if a user is downloading a huge file, it might be okay to put the monitor to sleep, but it's obviously not okay to put the network and/or HDD asleep which would sometimes happen due to odd "hiccups" in the download stream (heavy server load, lag, etc.). Thus, power management (PM) had to be moved away from the "usage model ignorant" BIOS to the more usage-model savvy and aware OS. ACPI is essentially PM done by the OS. It encapsulates standards that the OS, apps, drivers, and hardware must follow in order to do PM intelligently and elegantly throughout the system.

ACPI introduced concepts that allowed for OS-control of power management that included a new interpreted language for doing configuration and power management (ASL/AML), invocation of BIOS-supplied control methods to perform platform specific actions in a manner that was OS-friendly (e.g. multi-core, multi-thread) as well as formal definitions and semantics for CPU, system and device level low power states (C, S, D).

The framework is very robust, and it continues to advance to this day. For example, the CPU idle infrastructure has been used to innovate around CPU idle states with only minor changes since 1995 (when only C1-C3 existed), through the addition of SMT and multi-core products, to the addition of C states (CC1-CC6, PC1-PC10).

The ACPI has faced strong criticism among the OS experts. The creator of Linux, Linus Torvalds, has been a particularly harsh critic of ACPI: His view was shared by other OS experts as well. As a consequence, in 2009 Intel started the new Simple Firmware Interface (SFI), but this project never really took off.

<Intel comments>:
Linux uses ACPI on Intel Architecture PCs for configuration and power management.

On the other hand, there are some CPU centric technologies for tuning the active mode power of the chipsets through Intel's Dynamic Platform and Thermal Framework (DPTF) offering features such as configurable TDP (thermal design power) and LPM (low power mode). These features allow OEMs to customize their designs in regards to thermal and power/performance properties in the active mode, setting thermal limits or limiting the maximum power consumed by the main SoC when the device is in active use. In theory, DPTF can leverage ACPI-based information about devices, thermal zones, thermal sensors, thermal relationships etc., but also can work with information coming directly from drivers, giving the SoC manufacturers the option to hide all of these details in their proprietary device drivers.

<Intel Comments>:
DPTF serves a different purpose and is largely used to provide more customizable support of what ACPI terms Extended Thermal Model. It allows for advanced usages specifically in small-form factor devices to augment power and thermal management and works in conjunction with ACPI and all the aforementioned hardware-based power management. In many instances DPTF is used to calibrate and identify cross-thermal relationships to devices integrated into small form-factor devices, such that thermal hot spots may be mitigated by affecting power dissipation of surrounding components (e.g. reduce power of SoC due to proximity of a WLAN device that may be unable to dramatically reduce its own power). In many instances, setting thermal limits involves programming hardware-based power limits, and any hardware-driven thermal or power limiting actions is typically reported to an ACPI-aware OS. DPTF works in conjunction with ACPI and hardware-based fine-grained power management via Power Optimizer.

DPTF depends heavily on ACPI to both take input from the platform and provide feedback and controls to the platform. DPTF uses standard ACPI 5.0 objects such as _TRT (passive cooling) and _ACT (active or fan cooling) to provide transparency for thermal control. DPTF allows the OEM to expose control of configurable thermal design power (cTDP) and Low Power Mode (LPM) via the Windows OS DPPE settings that would be visible in the Advanced Power setting within the Windows control panel.

Many OEMs do not expose these controls directly to the end-user due to various hardware and thermal limitations or requiring specific additional changes in the platform to properly support the feature correctly. For example, configurable TDP (thermal design power) allows for 3 Thermal modes (up, nominal and down modes). These settings are associated with the cooling capability of the platform and are tied to different fan speed settings or modes of the PC such as docked or on AC power. Incorrect setting of these controls can cause spec violations that may impact the functionality of the product. Risk includes increased external device temperatures (Tskin) which may cause end-user burns or extreme heat and could lead to end-user harm and potentially expensive litigations to settle end-user claims. Additionally, if cTDP 'up mode' is used, device components may operate at higher than expected temperatures and damage the short term or long term reliability of the device. Additionally, some of these modes require specific power delivery modes such as AC or various battery sizes. Improper end-user selections may cause system shutdown, and could result in loss of end-user data. Finally, a typical user has very little knowledge of these settings and may inadvertently set Config TDP to a ‘low mode’ and experience very poor performance resulting in OEM brand value degradation, end-user service calls and complaints, which could potentially cost the OEM manufacturer millions of dollars in lost revenue and support calls. Microsoft has specifically limited the number of default power management configuration modes in later Windows releases due to large number of observed issues, caused by end-user inadvertently setting the wrong power management.

Security is also a large concern in the industry. We have already seen cases where hackers have used open power policies to degrade the system performance. Exposing and creating open standards and configurability, adds little value for the end-user, but greatly increases vulnerability for hackers to launch catastrophic global attacks on the power and thermals of platforms. Additional transparency, in most cases, could also drive up cost and complexity of the platform, software and silicon hardware, driven by implementing additional safeguards to prevent abusive attacks, focusing on the power and thermals of the platform. This could manifest itself with additional independent sensors on the platform, more hardware control logic, thermal protection circuits, etc. These additional costs to the industry are not only HW BOM cost but manufacturing cost due to extra validation and increased testing time. Thermal sensors must be calibrated and tuned for each unit and can substantially increase testing times.

DELL Notebook Power Management Analysis

In this project the research team explored the level of openness of the ACPI mechanism and level of ease for an OEM or smaller PC manufacturer to tune the power management of the system on chip, motherboard, device and the software in order to reduce the energy consumption of the product. The research team used the Dell Inspiron 5748 notebook to conduct the analysis and draw conclusions, selected based on its high ENERGY STAR reported idle wattage relative to other products.

<Dell comments>:

Author states that he selected the Inspiron 5748 due to its high energy star reported idle power but completely ignores true drivers of idle power in the system. This system has a 17 in HD+ display and external graphics which contribute to higher idle power and are completely unrelated to study findings regarding ACPI table population in the report. The authors completely ignore true drivers of idle power in the chosen system and no conclusions can be drawn about power management capabilities of volume notebooks by evaluating a single desk top replacement style notebook system.

The hardware features of the notebook’s main processing component are as follows:

- Intel i5-4210U chip includes two dies in one package:
  - Haswell CPU (22nm);
  - LynxPoint Chipset (32nm);
- Haswell CPU includes:
  - Dual-core CPU;

  - Graphics and Media accelerators;
  - Fully Integrated Voltage Regulator (FIVR);
    - Two voltage rails are supplied to i5 chip (6+ in IvyBridge):
      - one variable ~1.8V voltage rail supplied to the i5 chip;
      - one variable ~1.3V voltage rail for the memory controller;
  - Power Control Unit (PCU):
    - enables lower power C7 state even while display is on;
- Intel 8 Series PCH (LynxPoint-LP HM86):
  - Included in package of i5 chip (separate die);
  - PCH = Platform Control Hub;
  - PCH replaces Northbridge/Southbridge, features include:
    - PCI-Express Controller;
    - SATA HDD Controller;
    - USB Controller;
    - Gigabit Ethernet Controller;
    - DMA Engine;
    - SMBus/I2C;
    - HD Audio;
    - Serial Ports, Timers, Real Time Clock, GPIO;
  - 4 Voltage Rails: 1.05V, 1.5V, 1.8V, 3.3V;

In the Haswell datasheets, no information about the PCU and FIVR as key power management elements could be found. Specifically, the ACPI machine description tables (SSDT, DSDT) do not expose the critical building blocks needed to analyze or influence power management behavior. There were no power resource objects defined in the description, and the power state change methods (_PS0, _PS2, _PS3) were only defined for two major peripherals, the USB 3.0 and the Wi-Fi, as well as for a handful of the simple low-power peripherals, such as the serial interfaces. But even for those devices, the power control methods appear to be incomplete and were hence not revealing any useful information that would help optimize the power management behavior of a device. Therefore the example given in the ACPI specification v5.1 on page 58 for value-added hardware for placing the device in custom lower power states does not apply for this notebook. Given the fact that the Intel chipset offers many custom lower power states according to Intel's marketing material as well as many technical presentations about Haswell by Intel, we conclude that the ACPI methodology was not followed by Intel in order to control the power states of their devices.

<Intel comments>:
This is an incorrect conclusion. Inspection of ACPI-tables is absolutely the wrong way to identify power management inhibitors or assess the quality of the power management implementation. Intel recommends repeating analysis with the proper tools such as Battery Life Analyzer. It is not statistically significant to conclude anything from this report. Sampling a single system with high idle power likely due to features completely unrelated to what is being investigated provides virtually no information on the methodology or state of maturity of power management in the PC industry. Concluding from it that Intel does not follow or support ACPI specs on energy management is entirely baseless.

The Haswell architecture does support some Intel specific features for tuning of power and performance, namely configurable thermal design power (cTDP) and low power mode (LPM). These features can only be configured through Intel's proprietary DPTF driver which isn't publicly available for the analyzed Dell laptop.

<Intel comments>:

See our earlier comment in the other section. End-user configurability of cTDP or LPM has potentially **dangerous** consequences to the platform thermals and power delivery. DPTF exposes the means for the OEM to provide configurable options to the end-user via the Windows Power Plan Advanced Power Settings controls. However, the OEM must choose to expose the configurable option to the end-user, and provide the appropriate protections in their platform to prevent damage to their device and the end-user. DPTF does utilize ACPI and does provide a means for the OEM to define DPTF operation via ACPI. DPTF is a module and not all OEMs implement all portions of what DPTF offers. DPTF has also been open-sourced for operation on platforms such as Google Chromebooks and utilizes ACPI methods on those platforms. The conclusions on DPTF in this article are incorrect and are not properly researched likely due to the very limited amount of effort put into looking at various manufacturer implementations over a variety of systems. Single sample data is not representative of the industry implementation.

Our conclusion is that the power management technology resides in the Windows device driver DLLs from Intel and other chip vendors, without exposing power management controls in ACPI, which is the standard supposedly used in Windows for the power management framework to support the OEMs. While ACPI has provisions for declaring all power management details of devices in a description table, the notebook computer we have analyzed contains a very sparse ACPI description table, leaving most power management controls inside the closed source device drivers supplied by the chip vendors.

<Dell comments>:

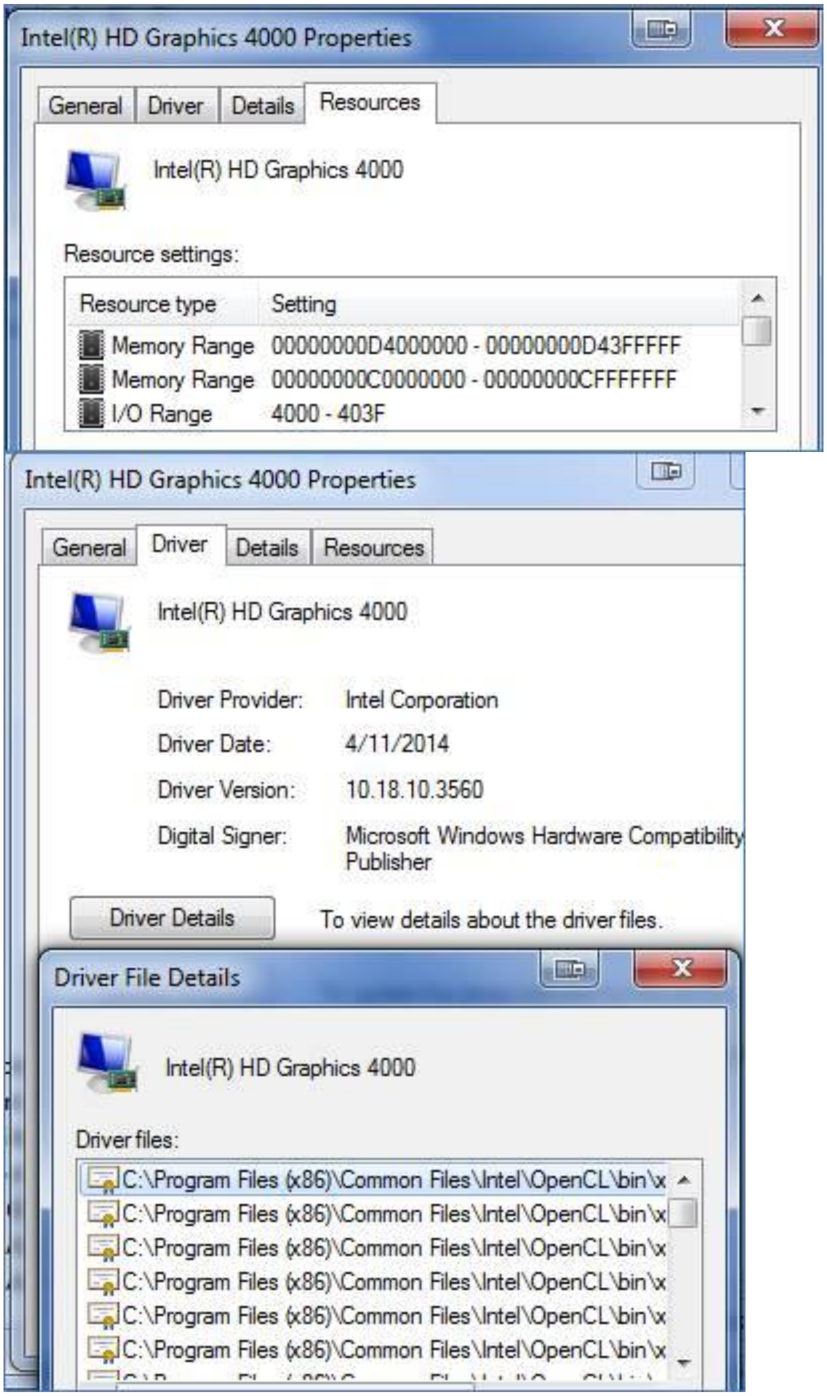
Power management in a PC system happens at many levels including OS, drivers, BIOS, firmware and hardware. Power management activities that happen at the device level in hardware and drivers are completely inappropriate to be exposed to end users as they do not have the knowledge or the experience necessary to make any meaningful decisions and are very likely to cause system operational failures, data loss and even melt the system or cause a fire. Yes experts with intimate knowledge of how devices actually work do the optimizations for their own devices in order to provide the best balance between performance and energy consumption while guaranteeing required functionality. There is no reason ACPI needs to be involved in every power management activity in every device in the platform and many power management decisions are completely inappropriate to be handled on the main processor due to excessive delay in decision making. Giving unknowledgeable end users control over these features would only provide them with the tools to reduce energy efficiency and potentially destroy their product as well as provide serious security vulnerability. The authors conclusion that there is something wrong about not exposing everything to end users or that the lack of disclosure of all power management capabilities in any way implies a lack of implementation of power management is itself completely wrong and shows a lack of understanding of the systems being evaluated.

On the application software side we are missing the ability to see how software components affect the power consumption of the device, especially when it comes to background tasks running while the PC is idle. There's no provision for software designers to declare or announce its resource requirements and it's also very difficult to observe resource usage for such processes, since Windows is running many background tasks as so-called services. These tasks don't get exposed in the system analysis tools provided by Windows or third parties. Instead, up to 16 services get run within a single service host process, of which multiple instances are running at any time. It is possible to monitor resource usage for individual processes, but for the service host processes there's no way to tell which one of the services is being executed at the time the resource usage is observed.

<Intel comments>:
Note: Capping maximum performance and power can also have drastic impact on energy consumption of a task. While limiting the maximum power can reduce measured wall power it may increase the time and resulting energy consumed to complete any given task. Even moderate demand workloads can benefit from allowing the platform and SOC to burst to a much higher performance for short periods of time. Abstracted controls in software or even end-user settings may actually increase the total energy consumed by the platform due to misunderstood and poorly set values and controls. Dedicated hardware and pre-configured controls that are set and not configurable, in many cases, create a more guaranteed user experience and power usage profile. Configurable settings, may in fact, allow the consumer to inadvertently increase the overall energy usage, as many of the energy controls could damage user experience, especially when the user perceives that just turning everything to maximum always guarantees the best experience. Educating the software developers and end-user consumer is a daunting task that could substantially increase the cost of PC platforms. Additional support for hardware, software and device vendors and handling end-user service calls and issues can be a large cost adder to the industry, and is not comprehended in this analysis.

Intel and Microsoft actively discuss better means to applications and services to improve energy consumption. While it is true that some APIs could and are being established for applications to expose its usage and resource demands, the value and viability of such an approach is questionable. It is relatively simple for a bad application to take advantage of the API infrastructure to abuse its power and resource allocation and starve the other well behaved applications. Enforcement and correction of bad applications is extremely costly. Other concerns around complexity due to security, hacking, and relative value to other efforts, remain as inhibitors to large scale development and adoption. Meanwhile the PC industry continues to reduce platform power at an amazing rate relative to other consumer devices.

A second problem on the software side is that Windows doesn't provide any transparency in terms of activity of its background tasks. Services are installed and activated without giving users any choice or information. It's difficult to figure out which application depends on which services, what those services are doing, how often they are being run, and when they are scheduled. The effects can be observed when monitoring resource usage and power consumption of laptop computers in their Short Idle and Long Idle states, which can sometimes be more active than when users are interacting with the device. Whether or not such activity is justified and whether the performed tasks are serving the user’s interest can only be determined through expert analysis, while the common user remains completely unaware of these activities and how they impact the power consumption of their device.



Resource type	Setting
Memory Range	00000000D4000000 - 00000000D43FFFFF
Memory Range	00000000C0000000 - 00000000CFFFFFFF
I/O Range	4000 - 403F

Driver Provider: Intel Corporation

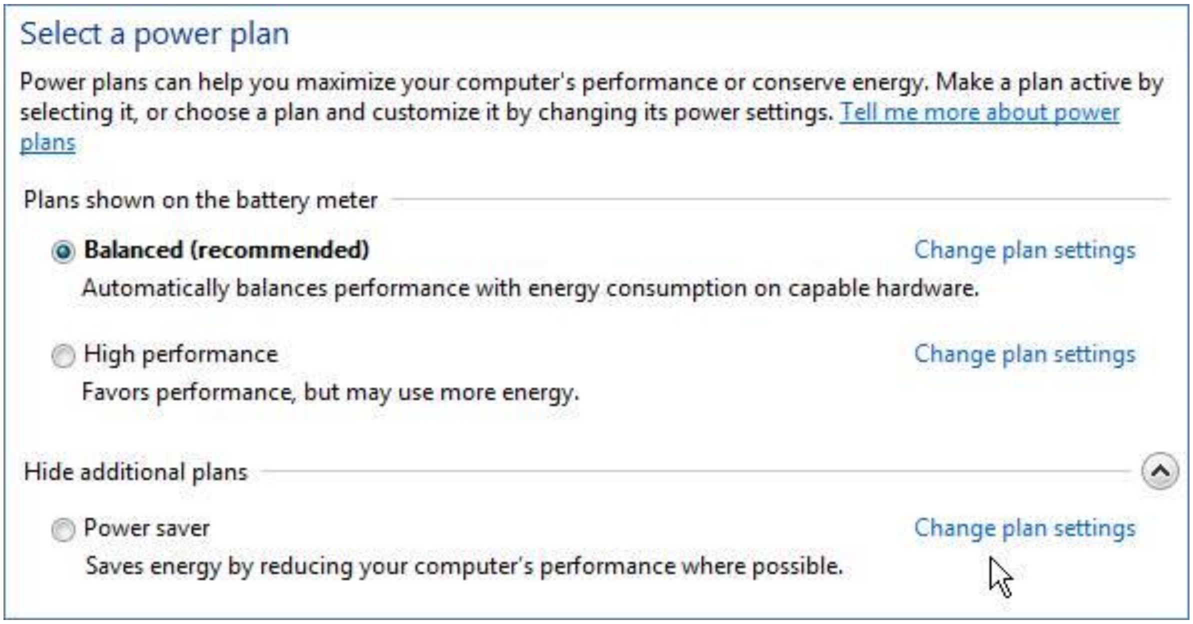
Driver Date: 4/11/2014

Driver Version: 10.18.10.3560

Digital Signer: Microsoft Windows Hardware Compatibility Publisher

To view details about the driver files.

C:\Program Files (x86)\Common Files\Intel\OpenCL\bin\x
C:\Program Files (x86)\Common Files\Intel\OpenCL\bin\x
C:\Program Files (x86)\Common Files\Intel\OpenCL\bin\x
C:\Program Files (x86)\Common Files\Intel\OpenCL\bin\x
C:\Program Files (x86)\Common Files\Intel\OpenCL\bin\x
C:\Program Files (x86)\Common Files\Intel\OpenCL\bin\x



Select a power plan

Power plans can help you maximize your computer's performance or conserve energy. Make a plan active by selecting it, or choose a plan and customize it by changing its power settings. Tell me more about power plans

Plans shown on the battery meter

● **Balanced (recommended)** — Change plan settings
Automatically balances performance with energy consumption on capable hardware.

○ High performance — Change plan settings
Favors performance, but may use more energy.

Hide additional plans

○ Power saver — Change plan settings
Saves energy by reducing your computer's performance where possible.

The lack of transparency and control on the hardware side as well as on the software side makes it difficult for OEMs to understand the impact of their design decisions on the energy consumption of their device, often leaving many optimization opportunities on the table. Especially newcomers and smaller vendors in this field will have a hard time achieving optimal energy efficiency for their devices, as they often don't have the means to receive the support of the large chip vendors for help in optimizing power and energy efficiency.

<Dell comments>:
Author should identify which major OEMs validated these statements that the OEM's cannot understand the impact of design decisions on energy consumption of their devices or that many optimization opportunities are left on the table as a result of what this extremely limited investigation finds. This entire paragraph is purely conjecture and author has provided no basis in fact upon which to make the statements.

Software developers often have no awareness of the energy impacts of their software. Even though some tools, such as Intel's VTune, are available to optimize the energy efficiency of their applications, as of today, little insight is offered for background tasks or services used by their applications16. This issue is exacerbated by the fact that software developers feel no pressure from their customers to focus on energy efficiency as users rarely associate poor energy efficiency with individual applications due to the fact that operating systems are lacking the infrastructure to break out energy consumption by software application More transparency and control could lead to more innovation in the field of energy efficiency, both by device manufacturers and software developers.

<Intel comments>:
The above is misleading because the software can and does request resources from the OS – memory and IO ranges, interrupts, time slice ranges, specific core usage, etc. Software can request power management events. Background software is present on a system, and discussions with concerned third party software vendors, take place regularly through Intel Software and

Services Group. In addition, for systems that supports Windows Connected Standby, the activity factor of background applications while in the Connected Standby state are actively managed to below certain thresholds. To a large degree, background apps are present for a reason (e.g. virus scanners) and they must be allowed to execute on events that matter to them. In general, drivers associated with hardware are highly optimized and subjected to rigorous testing for functionality as well as energy efficiency as part of any mainstream OS release. Again, VTune or Battery Life Analyzer can identify software that is problematic and what the bad behavior is (e.g. high frequency timer event injection, high periodic overhead, inhibition of device low power states, etc.). Intel's customers have tools to ensure their system ship without energy efficiency impediments.

Opening up the power management infrastructure

The results of this project highlights a systematic problem in the computer development chain exists: none of the parties in the design and production chain has the responsibility for the energy management and the comprehensive energy picture of the complete device including the main system on a chip (SoC), memory, components on the device, motherboard and the attached devices. The producers of the notebook SoC provide the reference design and the reference software including the OS drivers for the SoC. They are not responsible for the whole notebook device, as they do not know what kind of components their customers and their OEMs shall use for the subsequent motherboard development. The notebook manufacturers rely on the OS software including the drivers which the SoC provider is delivering. They mostly just take what the SoC manufacturer provides, so can manage the sleep states only through the OS system calls, not the OS or a dedicated energy management kernel what is the prescribed approach heavily used in mobile devices. This leaves the notebook motherboard with a number of power hungry components beyond the SoC which are hard to include in the power management, like the DRAM, flash memory, Ethernet controller, PHY, Wi-Fi, HDD and board level PMIC. Simply put, there is a systematic problem in the design and development of computers for the companies which are not large enough to receive a dedicated support service from the SoC and OS producers.

<Intel comments>:

Mobile devices, with so-called "open usage models", are STILL closed systems. The OS, the BIOS, the HW, and the "drivers" never change. PM solutions for these mobile devices can, and will be, totally different from what is required for open-architecture PC computers.

<Dell Comments>:

The author again describes how OEM's operate their businesses without any supporting evidence from major OEM's. Author should provide validation from those actually doing this work in the industry and stop making statements not founded in fact. All of the devices described above as being power hungry and hard to control have extensive power management capabilities with controls provided both to the OEM and in many cases to the end users where appropriate. The fact that these capabilities are not in ACPI tables in no way indicates their lack of existence.

This problem was supposed to be addressed by the ACPI technology presented in the previous sections. Unfortunately, the ACPI technology has barely changed during the last 18 years despite the rapid advanced in hardware and software, especially in the mobile space. As such, it is less and less likely that it can be of help to the OEMs or new innovative companies who are trying to reach the highest energy efficiency levels for their products. In the past, the BIOS (called also core system software), like Phoenix Technologies, American Megatrends and Insyde Software filled this void between the SoC and OS manufacturers on one side and the OEMs on the other by offering low level software and professional design services, which in particular were in high demand by the OEMs. As the activity of the BIOS was seen as an unnecessary cost factor, most of these companies are now exiting the BIOS business.

In the next sections we shall discuss approaches leading to a more efficient and widely deployable power management.

Energy Proportionality

For successful application and deployment of energy management techniques it is important to carefully analyze the application software activities, identify power state dependencies and objectively determine their realistic energy needs. Very often the lack of alignment between the various software programs and their development teams combined with the insufficient information about the power needs of the software causes all components to be kept at their full power all the time. A good example is the protocol software which provides the security, update, digital right management or communication synchronization between the device and the remote location during the low power modes.

<Dell comments>:

Author makes a patently false statement about state of system as a result of software. It is theoretically impossible to have all components at full power simultaneously and such a statement indicates a lack of honesty or understanding of the subject being discussed.
One example of a current 14 inch notebook computer has system hardware power at 4.7W at idle and 27W running a high performance workload. This seems to be an example of Energy Proportionality that the author states do not exist in the industry.

Such software should be executed in an energy proportional manner, which means that there is no reason to wake up the main processing cluster and all the peripherals of the notebook just to send and receive a limited number of internet protocol (IP) packets. In the current notebook designs not only is the energy intensive main processing cluster used for that purpose, but in order to be available at every possible instant for such an infrequent activity the whole device is kept up and prolongs entering into the deeper sleep states. Apple’s dark wake concept is a good example of the right approach to power management of such less frequent and background device activities. Energy proportionality should be the governing principle for energy management to guarantee that the work done by the software and the energy consumed by the hardware are all the time kept in the right proportions.

<Intel comments>:

For our wireless networking we have a technology called Intel remote wake in which the Wi-Fi NIC offloads sending network ‘keep alive’ packets, needed to maintain network connections while the host is in S3. This is similar to Apple’s dark wake and this work was all done to enable

S3 based solution but with Windows 8 and connected standby (CS), we focused on the CS based solutions in which the network (wired and wireless devices) maintain a connection in CS and waits for incoming packets to match a OS defined filter to allow various networking protocols to wake the host. This allows the platform to stay in CS and maintain networking application. A simple example is Skype support for this. With CS the Skype application sends a 'keep alive' packet every 30 seconds and up to 15 minutes to the Skype server to maintain the connection. The rate (30s – 15mins) is determined by the network type. While in CS the networking device listens and waits for packets to match the filter. So when someone calls you on Skype the incoming packet would match the filter and will wake the host from CS and the Skype app can ring the user. Additionally we support network scans offloaded in the Wi-Fi NIC so that only known Wi-Fi AP will wake the host. In summary we have the software and networking HW that in principal does the Energy proportionality.

Energy Design Standards

The key for transparency and industry collaboration in power management is the usage of standardized descriptions of the power and energy aspects of the system and the individual components. The industry needs a standard that can capture all of the complexities of today's hardware in terms of power states, dependencies between components, as well as the controls and conditions necessary for entering those power states. At the same time, the standard should also allow to describe the energy needs of the software running on those platforms. Software applications should declare the resource usage of their primary as well as background tasks in a specific manner, including intensity and frequency. Software should also expose controls to change how individual tasks are scheduled and how this affects functionality and quality of service.

<Intel comments>:

A single bad app can destroy the good behavior of all 99.9% of the good applications. What testing and compliance could be put in place and how is that going to enforced? What security standards could be in place? What additional HW and platform cost would be required to prevent intrusion and failsafe measures in the HW? This concept and idea is not new or innovative but faces all the same issues the industry has already recognized and explored on multiple occasions. The industry takes a measured approach on the value of exposing controls to the end-user based on risks discussed earlier. The proposed theory in practice is much more expensive and difficult to implement than the simple proposal of the theory. The industry, driven by market forces, has taken many steps to improve energy efficiency as demonstrated by the dramatic drops in platform power over the last 5-10 years. Enforcing proposed universal solutions in an industry, with a strong track record of energy efficiency improvements, seems counterproductive and potentially stifles the rapid development and innovation.

Currently, such standards do not exist. The energy design and management flow for electronic devices, including notebooks, is disconnected and lacks the abstractions, formats, interfaces, and automated methodologies long established and standardized in functional design and verification of hardware and software. The main disconnects (or collaboration walls) on energy issues are at the handover points between various design teams shown on Figure 3:

- From hardware designers and component providers to system integrators and power designers at the hardware level;
- From system integrators to firmware developers at the core system software level;
- From device developers and firmware developers to OS integrators at the OS API level;
- From device developers and OS integrators to the application programmers at the application software level;

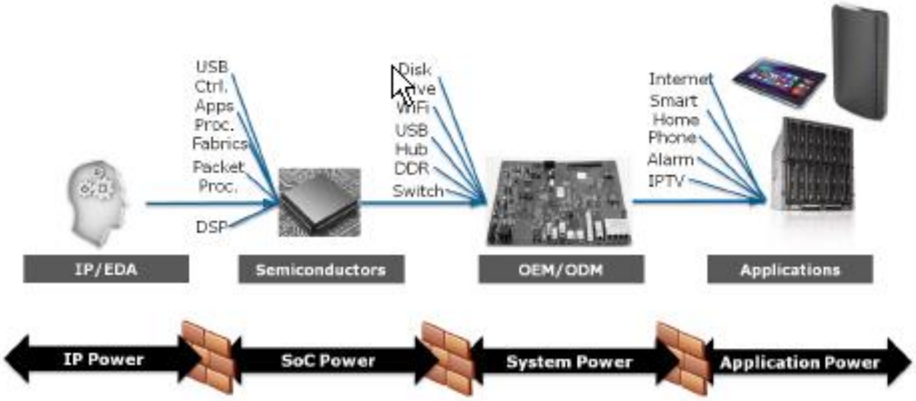


Figure 3: Power and Energy Design Flow

A new standard should address all of these challenges, mainly from an engineering perspective, by automating the design of power management software and hardware. But such an approach has broader benefits, including:

- Lowers the cost and shortens the time-to-market, as companies across equipment classes can share the same power management infrastructure;

<Intel comments>:

In many instances the standards could increase costs to maintain compatibility with existing older “standard” infrastructure.

- Enables the introduction of formal energy description of equipment and automated generation of the power management software and hardware;
- Fosters energy savings competition among manufacturers of various equipment classes;
- Allows for unified run-time reporting and control of equipment's energy consumption;

<Intel comments>:

This opens up large concerns on security, exposing industry to large scale hacker assaults, misuse of equipment or even damage due to incorrect settings and potentially expensive litigation to resolve field failure or end-user complaints or physical harm.

- Establishes tighter minimum energy efficiency requirements for higher energy savings and enables faster adoption of energy standards for future equipment classes;
- Allows unified testing, measurement, and standardization procedures; Brings together a larger pool of power management experts to focus on the same problem and improves technical education in academia;

In August 2014 the IEEE Standards Association formed a new IEEE standardization project and the IEEE P2415 Working Group (WG) focused on the “Unified Hardware Abstraction and Layer for Energy Proportional Electronic Systems”. The IEEE P2415 WG shall standardize the best energy savings practices from the electronic industry and make them available to the public. The P2415 WG includes a large number of worldwide leading electronic companies who supported the P2415 initiative during the 12-month formation process, including ARM, Atrenta, Broadcom, Cadence, Cisco, Comcast, Doceapower, IBM, Intel, LGE, Mentor, Microsoft, Pace, Qualcomm, Realintent, Synopsys and Xilinx.

Software and Tools for Energy Proportional Design

To fully leverage the energy standard for energy proportional devices, it is advisable to develop software tools acting as the design, visualization, generation and test/measurement cockpit for all energy management and design related activities. The required features of such tools are:

- Create and edit descriptions of the hardware and software details of their systems;
- Export and import a variety of system and power description formats, including ACPI, SFI, DTS, UPF/CPF and IP-XACT;
- Represent the device and component hierarchy in graphical and tabular format;
- Enter and represent the FSMs as state transition tables (STT) or diagrams;
- Verify the validity and consistency of the description;
- Create and manage the IEEE P2415 library of components and devices;
- Generate the energy management driver, user space daemon and kernel for always on cores;
- Generate the FSM for dedicated hardware implementations;
- Simulate power management scenarios on FPGAs or virtual prototypes;

The tool should be integrated with physical power measurement and data acquisition (DAQs) devices for automated measurement, test and verification of the power and energy data collected from multiple voltage rails, supply inputs and battery contacts.

The run-time energy management software should run on the device and complement the standard OS power management. The main tasks of such software are:

- Detect and respond to power management directives by the OS or during OS transitions;
- During OS suspend periods manages the sleep, suspend and dark wake states of the device;
- Coordinate power state changes of components, clusters and subsystems;
- Manage IEE P2415 operating points and scenes;
- Execute control code to retain state information during transitions;

- Change power states by directly accessing hardware components, including clock dividers, PLLs and PMICs;
- Provide run-time power and energy estimates;

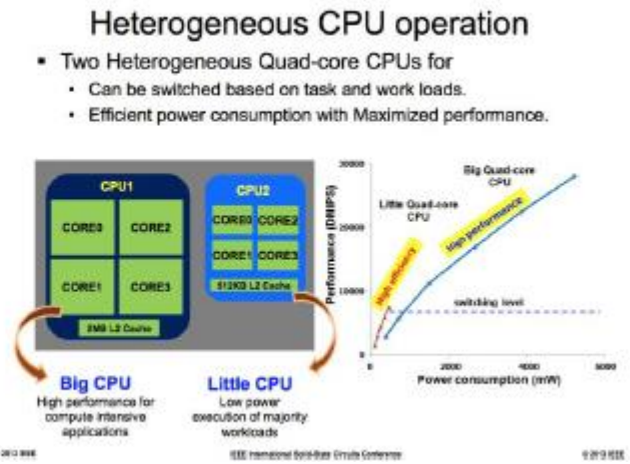
The energy management software has to have:

- Fast and precise execution of power state changes;
- Small footprint to fit within local memory of dedicated cores;
- Flexibility to distribute energy management across all the participating cores and the OS(es);
- Support for wide range of processors ranging from application processors to dedicated power management and always-on-cores;

Energy-Driven Component Hierarchy

Energy proportional systems require a shift towards energy proportional components which scale their energy consumption based on the workload. Some components exhibit better energy proportionality than others. The energy consumption of processors can be scaled by scaling the clock frequency, the voltage or just by cutting the voltage supply altogether. The memory can be put into a low power mode (refresh mode) without losing its content or turned off in which case the memory content gets lost and has to be preserved by the software. Communication components like Ethernet or Wi-Fi are harder to scale as their energy consumption is dependent on interactions with other electronic components mostly outside of the motherboard and the device.

The mobile industry is increasingly introducing component hierarchies where the same internal functionality is provided by a number of alternative components with different energy characteristics. Similar to memory hierarchies with the L1 cache, L2 cache and the external memory, where the tradeoff is between speed, area and costs, mobile devices increasingly include processor hierarchies in order to smoothly control the energy consumption. An example of such a processor hierarchy is presented in an Anandtech article (Figure 4).



Heterogeneous CPU operation

- Two Heterogeneous Quad-core CPUs for
  - Can be switched based on task and work loads.
  - Efficient power consumption with Maximized performance.

Figure 4: Processor Hierarchy for Energy Proportionality

Improved power management in notebooks can be achieved by adding specialized microcontrollers to the device, also called always on cores or power management units. Such cores execute most of their software from private memories and consume between 50 and 100mW, which is one or two orders of magnitude less than what the powerful and energy hungry main processors used in notebooks consume. Although such microcontrollers are part of almost all PC systems on chips found in desktops and notebooks, their use today is limited to the operations within the system on chip and are

not used for controlling the energy activities of the motherboard and the whole device. Even more importantly, no provisions exist for the support of such microcontrollers in the operating system, so its use is limited to the system on chip producer and excludes the OEM.

Cost/Benefit Analysis

The hardware and software modifications and the engineering design procedures necessary to reach the savings in short idle energy consumption of 25% and long idle of 50% were estimated based on the qualitative energy analysis using the available data sheets, as well as the results we have achieved on the IPTV set top box project. (In order to provide a detailed quantitative analysis of the proposed improvements for computers we recommend a project to develop a fully functional computer reference design and to conduct detailed measurements, testing and costs/benefits analysis. The estimated duration of such a project is between 12 and 18 months.

<Intel comments>:
Intel recommends that CEC spend the 12-18 months to do this characterization on a proper PC system using state-of the art tools. It would be better if the CEC commissioned a non 3rd party (one without any biases regarding the solution space) to perform this study.

<Dell Comments>:
Entire cost benefit analysis is based off of pure conjecture with no basis in fact or established correlation between projected improvements/costs and PC's currently shipping into California. Author never states how data sheets were used to establish a system idle power level reduction. The only data sheets referenced were the Intel processor data sheet and that data sheet does not give energy use consumption information. Even if processor energy consumption information is obtained no relationship was established between these data sheets and system level energy consumption of PC's in California market.
Absolutely no correlation was ever established between improvements made in set top boxes and the current state of PC's in California.
Projected power level reductions are pure conjecture by the author. Author also fails to define the time line to get this level of changes into production systems which is most likely to be on the order of eight to ten years.

The additional costs for the proposed improvements fall into three categories: 1) component and royalty costs per device, 2) annual costs for the design technology and software infrastructure and 3) annual engineering labor costs.
For the costs for the inclusion of an always-on microcontroller on the SoC and to open up the energy management software we estimate at 15 cents per computer. These costs are based on the

typical royalty costs for this type of microcontrollers and the accompanying energy management software.

The cost for the increase in the silicon area of the system on chip is minimal and was neglected.

For the engineering team developing the improved computer the necessary investment for the software and tools infrastructure, the estimate is at around $1M per year. This includes the term (annual) licenses for various electronic design automation (EDA) software tools for energy design and management and the run-time software. Similar to other software term licenses, it represents the part of the costs for the software which is provided by specialized third parties and shared among the involved companies who use it.

For the engineering R&D labor to implement the hardware and software improvements, we estimate $2M for the increased non-recurring engineering (NRE) costs per year. We assume that these costs shall continue year after year as long as the company is in the computer business and is devoted to produce energy efficient devices. It is predicted that the reuse effect on the amount of work in the subsequent years shall be offset by the increasing complexity of the new components, software and devices, as well as the increased efforts to sustain the energy gains.

On the 59 million new computers (desktops and notebooks) shipped annually in the US and estimating around 10 computer engineering teams dealing with the hardware and software for energy optimizations, the software/tools and labor costs results in an additional NRE cost of approximately 51 cents per new computer produced. Summarizing the above we estimate that the total additional net costs per new computer shall be not more than 70 cents and have a retail price of less than $1 with the incremental retail markup as per the 2012 estimates of the Department of Energy for battery chargers. These costs are well below the typical CPU price of $40-$164. Utilizing duty cycles taken from Appendix B and power draw estimates taken from the ENERGY STAR 6.0 QPL (October 2014), the energy benefits are calculated as follows:

- Desktops spend ~20% of their duty cycle in long idle (i.e., 1,753h per year) consuming on average 25.2W and ~20% in short idle consuming on average 26.8W;
- Notebooks spend 10% in long idle (i.e., 876h per year) consuming on average 4.85W and 10% in short idle consuming on average 8.55W;

Based on the above numbers, which are likely an underestimation of the market average given that ENERGY STAR products are the most efficient and assuming the current rate of 15 cents per kWh, a typical desktop computer with the suggested improvements would save 34 kWh/year and a typical notebook 4 kWh/year. Assuming a 4 year lifetime for desktops and a 3 year lifetime for notebooks, the net benefits of ~$20 and ~$2, for desktops and notebooks respectively, exceed the incremental costs described above, with lifecycle benefit cost ratios of ~22 and ~2, respectively.

Assuming a 37-63% split between produced desktops and notebooks, the suggested improvements would result in the combined savings for the US electricity ratepayers of $133M in the first year and ramping to $510M in the 4th year. This compares to the total incremental costs of $54M per year.

<Intel comments>:
For reference, the above assumptions translate into:

	Desktops				Notebooks			
Modes	Hours/Yr	Duty	Power	kWh	Hours/Yr	Duty	Power	kWh
long idle (W)	1753	0.2	25.2	44.17	876	0.1	4.85	4.249
short idle (W)	1753	0.2	26.8	46.98	876	0.1	8.55	7.490
Annual Idle (kWh)				91.1				11.738
Target kWh Savings				34.				4.000
% kWh Reduction				37%				34%
Target Power to achieve kWh savings								
long idle (W)			15.80				3.04	
short idle (W)			16.80				5.36	

<Intel comments>:

These costs ignore the collective investment that the industry is making to achieve tangible end-user meaningful power savings. The industry continues to invest in low power displays, solid-state storage, low power memory, and system-level integration of various platform ICs into a single SoC. Overlooking these larger issues at hand implies that an opportunity cost analysis was not performed to determine where is the best investment to make to improve energy efficiency. “10 computer engineering” teams vastly underestimates the size of the industry and the various value-add that occurs in each step of the design, validation, integration and production of a PC. The PC industry consists of dozens of MNC OEMs and local OEMs plus the channel which consists of thousands of small-scale system integrators and individual users. The diverse ecosystems of system builders have varying degrees of ability to validate each system’s HW & SW BOM for power savings features. Due to our diverse ecosystem, we suggest that CEC take a measured approach that is challenging without impeding innovation, and consumer choice.
We don’t see how the proposed software based mechanism can deliver ~10W of power savings in a desktop system. The power savings opportunities predominately come from the selection of lower power hardware which is generally more costly than the estimated $1/system.

Conclusion

The qualitative analysis presented in this report proposes the steps necessary to improve computer energy consumption. The report suggests opening up the power management hardware and software in order to improve the collaboration on energy issues between various companies and design teams developing the computers. The largest potential for improvement is in the area of energy aware software development where the improved energy proportionality can lead to major additional savings. To reach such savings we advocate investments in industry standards, software, tools and development of energy proportional components. Our cost/benefit analysis indicates that every dollar invested in such improvements has the potential to return almost ten dollars in energy savings for the ratepayers year after year. In order to provide a detailed quantitative analysis of the proposed improvements we recommend initiating the development of a fully functional computer reference design and conducting detailed measurements, testing and costs/benefits analysis.

<Intel comments>:
Much of the work in Appendix C is speculative and lacks understanding of ground realities, and by Aggios' own statement on page C-27 first para, “The hardware and software modifications and engineering design procedures necessary to reach the savings in the short idle consumption of 25% and long idle of 50% were estimated based on qualitative energy analysis using the available data sheets, as well the results were achieved on the IPTV set top project”.

There is a fundamental difference between the set top boxes and the general purpose PC’s with respect to power management. Set Top Boxes are single purpose device, as such, they do *one* thing. Our key concern with this proposal is that data that is being used is based on a closed, single function device (IPTV set top box device). It is orders of magnitude easier to implement power management solutions on a closed, single-function device, than it is to implement a similar solution on an open, fully-configurable device such as a consumer PC. These numbers are created by extrapolating from a set-top box (single purpose compute device). We believe the analysis and conclusions of this paper are simply unfounded.

Industry welcomes the opportunity to further educate CEC and other stakeholders to clear misperceptions resulting from Aggios' report. CEC’s heavy reliance on this report for its staff draft proposal is a serious concern to us, and we are committed to work with CEC to set the record straight.

APPENDIX 1

ACPI Overview

With the advent of open systems like the IBM PC came the need for OS discovery of the hardware components that required software support as well as the support for the dynamic connection and removal of peripherals. With the advent of Laptop systems this included connecting to a peripheral dock. Along with the advent of mobile computing came the need to conserve power as much as possible to deliver adequate battery life.

Several solutions emerged for these needs. Systems moved from Static configuration to dynamic configuration utilizing fledgling Plug-and-Play specifications. On the power management side, the Advanced Power Management (APM) specification emerged, which allowed the OS to communicate with the platform that was idle, and the platform to perform power management.

Unfortunately, neither of the solutions was robust enough to accommodate growing platform needs. Plug-and-play was nicknamed Plug-and-Pray. APM had a few drawbacks including no direct OS interaction with or awareness of peripheral power management coupled with the OS handing off operation to firmware running native code.

As a result of this, by the mid 1990’s PC systems had a number of user experience issues that were adversely impacting the business. These included systems that were not always working with their attached hardware, systems that had worse power management vs. others, confusing and different means for users to configure power management settings (bios setup, OEM custom apps, etc.). Further it manifested in OS vendor’s inability to deliver a stable, high-quality, OS product as a result of not being able to validate the OS across the plethora of systems that could support the OS but at the same time contained native code, the OS called in their firmware. Lastly, there was the quintessential problem of the platform performing power mgmt. while the system was doing something from an application perspective that should have impeded certain power management actions. E.g. the screen should not go blank during a presentation or during video playback.

To address these problems, Intel, Microsoft, and Toshiba formed the SIG to develop and Open Industry Specification known as the Advanced Configuration and Power Management Specification (ACPI)
ACPI’s goals were multi-faceted. These included:

- Defining a standard hardware / software interface that the OS could use to perform power management (moving power management decisions from the platform to the OS).
- Providing a mechanism to allow the platform to perform platform specific functions as necessary in support of both dynamic configuration and power mgmt. while eliminating OS invocation on native platform code.
- Establishing an interface that allowed the OS to discover the platforms memory configuration, devices and topology so that it can dynamically find and load the appropriate SW drivers for the HW while also understanding the relationship between

devices i.e. and inter-device dependencies and the routing of interrupt sources and resource requirements of devices so that the OS can rebalance resource assignments as necessary.

- Subsume the processor configuration information provided by the MPS specification.
- Defining hardware / software interfaces that enabled the OS to perform Thermal management
- Defining an overall power management framework and nomenclature for both system and device power management.

Moreover, ACPI’s goal was to provide mechanisms to do things when no other native mechanism existed. For example, if a device could power manage it then let it. If a device could not power manage by itself but there was a platform control that existed to power down an unused device, then it should allow the OS to power down that device when it is not used by invoking the platform control. Similarly, configuration changes like hot-plug could also be handled using ACPI defined interfaces if the capability was not natively supported by a bus protocol.

At the same time, an ACPI aware OS provided new UIs and interfaces that allowed the end-user to control power management in a common way among systems and allowed application to interface with power management policy to achieve desirable results, e.g. not turning off the display when it was required. In the Windows OS these capabilities were known as ‘OnNow’, which was comprised of ACPI support + the new WDM driver model that supported the ACPI framework and win32 API extensions for power mgt.

Systems that supported the new ACPI model were more reliable and delivered more consistent power management as well initiating the trend towards all PCs turning off or minimizing power when possible while quickly returning to full operation when necessary/needed.

It is important to note that while ACPI defined a power management. framework and various system and device state definitions; great accommodation was made for OEMs and device manufacturers to differentiate within the framework e.g. the means to dynamically change power levels within an ACPI power state was not precluded. The best of example of this is Intel’s Converged Platform Power Management (CPPM) solution, which delivered substantial reduction in run time system power while operating within the ACPI framework.

The elimination of the OS calling Native platform code was achieved via the definition of an interpreted code interface that provided limited but all necessary functions to perform both platform configuration and power management.