

White Paper
Mudit Vats
Graphic Software
Engineer
Intel Corporation

Ishu Verma
Software Technical
Marketing Engineer
Intel Corporation.

Linux Power Management

IEGD Considerations

March, 2010



Executive Summary

Implementing Power Management (PM) is a complex undertaking involving seamless interaction among several hardware and software components. PM in Linux is even trickier as the open source choice enables permutation of various PM players and solutions for a given system.

Troubleshooting display-related issues when resuming from S3 suspend state in Linux is an extremely complex endeavor, requiring an understanding of the handoff among various hardware and software components.

When a system resumes from a power state, such as S3, handoffs among several variables (BIOS, VBIOS, kernel, drivers, windows manager and X) put the system back to its previous state. This white paper focuses on the blank display (or no display) issue related to power state S3 suspend/resume.

With no display after resuming from S3 suspend state, the graphics driver may seem to be the culprit. However, this behavior is often the result of bugs in device drivers, BIOS, or VBIOS.

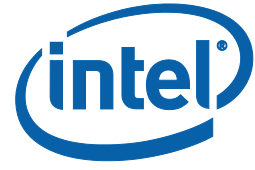
This paper presents some of the common techniques that you can use to discover the root cause of the blank display issue, among them booting headless, using an external graphics card, and using an open Linux graphics driver. Some possible solutions involve disabling video repost and optimizing `xorg.conf`.

The Intel® Embedded Design Center provides qualified developers with web-based access to technical resources. Access Intel Confidential design materials, step-by step guidance, application reference solutions, training, Intel's tool loaner program, and connect with an e-help desk and the embedded community. Design Fast. Design Smart. Get started today. www.intel.com/embedded/edc. §



Contents

Power Management and Linux	4
Power States	4
S States (System States)	4
D States (Device States)	5
Software Technologies	5
PM Framework	5
X Windows System	6
Window Managers	6
PM Implementation with IEGD	7
Suspend to RAM	9
Resume/Wake-up	11
What If Things Go Wrong?	12
System BIOS Issues	12
Video BIOS	12
Miscellaneous Issues	13
Diagnosing PM Issues	13
Conclusion	15
Glossary	16
References	17



Power Management and Linux

Power management (PM) is a key feature for most computing platforms. PM provides various benefits: increasing battery life; lowering heat emission; lowering carbon footprint; and increasing the longevity of devices such as panels, hard disk drives, and internal components. Implementing power management is a complex undertaking involving seamless interaction among several hardware and software components.

Before implementing PM, it is important to understand the available hardware support for the various devices, among them CPU, DRAM, Flash, hard drive, radio, Bluetooth*/WiFi/3G, and display.

Some examples of hardware PM include:

- CPU: HLT, Stop Clock, SpeedStep®
- Display: Blanking, dimming, Energy Star, power-save mode.
- Graphics Controller: Powering down (completely off or into an intermediate power state).
- Hard Drive/CD-ROM: Spin down
- Network/NIC: Wake on LAN
- USB: Device power state transitions (mouse, USB drives, speakers); Wake on access (mouse movement, R/W access, input signal)
- BIOS: ACPI S3/S4 platform behaviors, video repost, CPU PM settings (states enabled, CPU fan throttling), and platform settings (e.g., thermal low/high watermarks, chassis fan throttling).

Power States

There are various power states at the component and system level to enable power savings. The following tables describe S States and D States in Linux.

S States (System States)

“S” States are responsible for platform-wide power state transitions.

**Table 1. System States**

State	Description
S1	CPU stops executing, cache is flushed; power to CPU and RAM is maintained. Devices that can power down are powered down. Older machines tend to support S1 versus S3.
S2	A deeper sleep. CPU is powered off. Not commonly implemented.
S3	“Standby” in Linux. “Sleep” in other operating systems. Also known as “Suspend to RAM”.
S4	“Hibernation” in Windows. Also known as “Suspend to disk”.
S5	Soft power off. The system is off, but some power is available for wake-up events (e.g., wake-on-LAN).

D States (Device States)

“D” States are responsible for device-specific power states transitions.

Table 2. Device States

State	Description
D0	Device is fully on.
D1	Intermediate state defined by device.
D2	Intermediate state defined by device.
D3	Device is powered off.

Software Technologies

The following sections describe various Linux software technologies that implement some part of power management.

PM Framework

Linux supports two implementations of firmware power management: Advanced Power Management (APM) and Advanced Configuration and Power Interface (ACPI). APM is an older power management technology focused on basic system and OS power management, with much of the power management policy in the BIOS. It requires an APM driver as the interface between the BIOS and the Linux operating system. With this method, power management events could pass between the BIOS and the OS. Devices can be notified of these events and respond appropriately. APM is still supported in the kernel, but most newer platforms support ACPI, the successor to APM.



The Linux kernel has supported ACPI for a long time. ACPI provides greater power management state support, platform configuration support, and allows for platform independence and OS control over power management events. In those ways, it differs from APM, where BIOS was mostly in charge of power management policies. ACPI supports power management, but also thermal (fans), buttons/lids (events), CPU “C” states, and power source (battery, AC), etc.

X Windows System

The X Window System provides graphics capabilities to the operating system. It provides an infrastructure which allows Window Managers to provide a desktop, widgets/controls, application windows, and toolbars, etc. The X Window System has existed for many years and has been ported to multiple operating systems. In terms of power management, the X Windows System provides support for both APM and ACPI. For APM, it supports several of the APM states. There's also support for User/System Standby, User/System Suspend, User/System Resume, and “Critical Standby/Suspend/Resume.” When a power management event occurs, the X-Server is specifically notified of system power management events. So, in this case, the X-Server implements what to do in the case of power management events.

In terms of ACPI support, the X Windows System supports ACPI graphics messages through acpid, but not system-wide power management events. Because the X Window System is a user process, if the kernel mode driver handles the suspend/resume power management correctly, the user mode process of the X Window System should work like any other application when put into and out of a power-managed state. To help facilitate state transitions, the Enter/LeaveVT functions can be used to save or restore a graphics state when transitioning from graphics, to console, and then to suspend. The reverse is true as well when going from resume to console, and then back into X.

Window Managers

As mentioned previously, Window Managers provide the graphical user interface with which a user interacts. Linux supports many different window managers. Two of the more popular window managers are GNOME and KDE. These window managers have quite a bit of development support and many distributions validate their distributions with these windowing solutions.

GNOME supports power management through the GNOME Power Management solution, which uses hardware abstraction layer (HAL) open source platform management, built on DBUS (open source messaging interface). KDE3 supports KPowerSave, its own power management solution. Both of these window managers attempt to accomplish the same thing: deliver reliable power management of the OS platform. OS vendors integrate other third-party, open source technologies and include them in the distributions, including scripts to ensure the correct sequence of events



happen to successfully go into and out of a power managed state. Depending on the solution, windows managers also may include a set of APIs that can be used to hook into a particular power management solution.

Aside from these power management solutions for Linux, there are many other “home grown” power management implementations, new windows managers implementing even newer solutions, and several old and new technologies. For example SWUSP, TuxOnIce, /sys/power/state, /proc/acpi/event, and others, implement some part of Linux Power Management. Having many power management players and solutions for Linux is good. The complexity of it all is that “there are many power management players and solutions for Linux”.

PM Implementation with IEGD

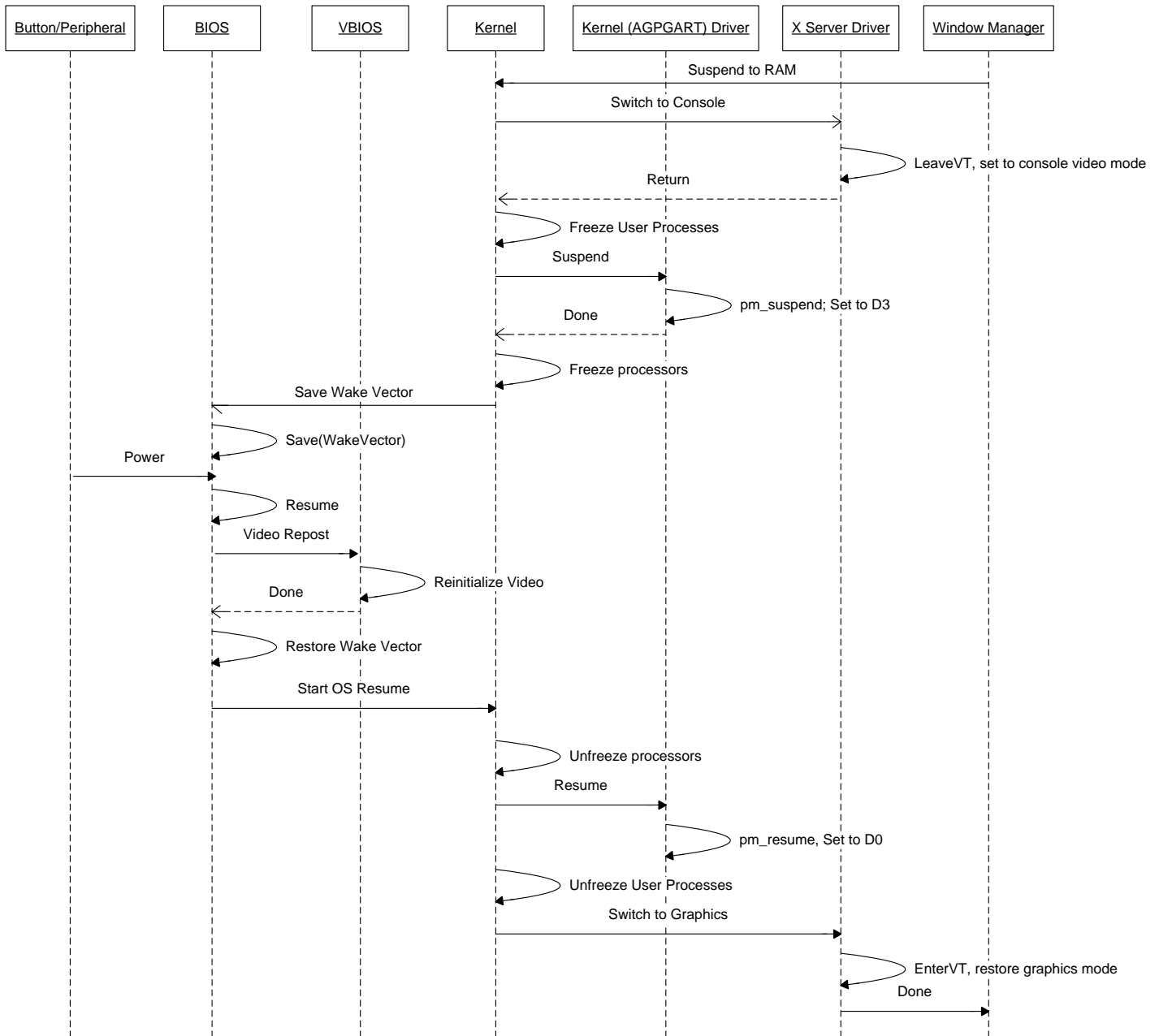
PM software manages state transitions along with device drivers and applications. Device drivers are responsible for saving device states before putting them into their low power states and then restoring the device state when the system becomes active. Generally, applications are not involved in power management state transitions. A few specialized applications that deal directly with some devices may want handle state transitions. This section explains the sequence of steps the Intel® Embedded Graphics Driver (IEGD) for Linux goes through during power management state transitions.

In Figure 1. Power Management Sequence Diagram, the sequence can be read by first understanding the top level blocks and then considering the events that occur over time from top to bottom. The blocks are described as:

- Button/Peripheral: Describes input from a physical device such as a button, key-press or mouse movement.
- BIOS: System BIOS responsible for platform configuration.
- VBIOS: Video BIOS in charge of pre-OS graphics initialization.
- Kernel: The Linux Kernel.
- Kernel (AGPGART) Driver: IEGD IKM (Intel Kernel Module) responsible for graphics device initialization and resource allocation.
- X-Server Driver: The X Windows System graphics driver for an Intel graphics device.
- Window Manager: The Graphical User Interface.



Figure 1. Power Management Sequence Diagram



The following two sections correspond to the sequence diagram and describe Suspend to RAM and Resume/Wake-up.



Suspend to RAM

When the system suspends or resumes from a power state such as S3, there is handoff among several variables (BIOS, VBIOS, kernel, drivers, windows manager and X) to put the system back to its previous state.

Examine what happens when the user puts the system into S3 suspend mode, i.e., Suspend to RAM.

Note: Each distribution may have its own way of putting the system into a suspend mode using any previously mentioned power management technologies available in Linux.

The Suspend to RAM action starts when an “event” is triggered in the platform telling the OS that a power management event has occurred. The event can be in the form of a button press, such as holding down the power (or sleep) button on a laptop to trigger a sleep action or it can be through user interaction, such as clicking on a window manager option to put the platform into a suspended state. Certainly, one can execute a script at the terminal or have a timer setup to automatically put the system in a low power state as well. In the example above, assume the user chose a window manager option to signal the GUI to change to a lower power state: Suspend to RAM.

When this action occurs, the window manager tells the kernel to trigger an S3 power management operation. How this happens is distribution- and implementation-specific. Newer Linux distributions have a platform management infrastructure that provides a protocol for the OS to communicate system or platform events between the software components and the kernel. By using such an infrastructure, power management events can be broadcast to other parts of the system through the kernel. Figure 1 shows that the window manager, after detecting the user-generated event, triggers the kernel to go to a lower power state. The kernel then starts the suspend to RAM procedure.

Alternatively, the suspend action triggers a hardware interrupt that is handled by the embedded controller. The embedded controller triggers a register in the southbridge chipset, setting a flag that a general purpose event has just occurred. The OS notices this, and checks the DSDT for what must happen next. Normally, this just calls a notification event that returns to user space via `/proc/acpi/events`; the user space software determines what happens next. Finally, the string “mem” is written to `/sys/power/state`.

The X display must switch to console before going into a lower power state. With ACPI, the Linux kernel first switches to console mode and then continues with its power management code. This is in contrast to APM where suspend/resume often occurs through messages that were trapped from within the X-Server. When X switches to the console, it calls a `LeaveVT` (“leave virtual terminal”) function that allows the switch to occur. This is where IEGD saves the graphics states.



The process saves all graphics registers and state information so that on resume, it can restore the graphics mode and state. After saving all of the registers, the process restores the previous mode—the mode before the OS entered into the GUI, i.e., console/terminal display mode. After restoring the previous mode, the end-user sees a text mode display on the screen and the kernel/OS can display text, which helps when debugging power management issues.

After restoring the display to the console, the Linux kernel freezes all user processes. This includes the X process and any other application running on the system (window manager, browsers, X applications, etc.). With the user applications frozen, the kernel can work with each device to ensure those devices support the power management operation correctly and then invoke the Suspend to RAM operation. After the user processes freeze, the kernel probes each device driver and calls its associated suspend function (if implemented). This gives each driver the opportunity to flush queues, stop interrupts, process any remaining interrupts, and lower the power state of the device. Some devices may or may not support putting the device into any other lower power state other than D3 (powered off). Other devices may support intermediate states, such as D1 or D2, i.e., the resulting power state depends on the device capabilities.

In the case of IEGD, the driver puts the device into D3 no matter what kind of power management event occurs. However, the driver sets D3 behavior in the graphics controller by turning off the graphics device's clocks on D3. On a Suspend to RAM that makes a device operate in a lower power state.

After all devices have been put into a lower power state, the only code left running is the kernel. The kernel does its own clean-up and freezes all processors except for the one currently running. After running the kernel-side suspend code, a couple of ACPI methods are executed: PTS (Prepare To Sleep) and GTS (Going To Sleep). These executions tend to poke various things that the kernel knows nothing about, and so a certain amount of "magic" may be involved. At this point, the system should be dormant. Next, the address of the kernel wakeup code is written to a location in the Fixed ACPI Description Table (FADT). Finally, DSDT values are written to registers described in the FADT. This usually causes some sort of system management trap to ensure that the memory starts in self-refresh mode; it also sequences the machine into suspend. For the S3 power state, this involves shutting the machine down completely except for the RAM.

With the wake vector set, and user processes, devices drivers, and kernel all in a "cleaned-up" state, the BIOS can now put the platform into a lower power state.

Note: There is usually a setting in the BIOS that indicates what action to take for a suspend operation. The options usually vary from a more aggressive power-managed state to less aggressive power-managed state. In that case the OS instructs the BIOS that a lower power state is desired. The BIOS often contains the policy for a lower power state.



Resume/Wake-up

When the user takes an action to wake the system, for example, a keystroke, mouse movement, or a power button press, the system switches on. It then jumps to the BIOS start address, does some setup such as programming the memory controller, and then looks at the ACPI status register.

Note: Some types of Video BIOS support the ability of “video repost.” This allows the Video BIOS to be called by the BIOS at the time of a resume operation (providing the BIOS has support for this feature). Video repost is the full re-execution of the video BIOS code. This helps when the graphics device is not working properly at the time of resume.

The ACPI status register indicates that the machine was previously suspended to RAM, so it then jumps to the wakeup address programmed earlier. The wakeup address leads to real-mode x86 code provided by the kernel, which programs the CPU back into protected mode and restores register state. At this point, the kernel code continues execution.

From this point, the process is essentially the reverse of the suspend process. The ACPI WAK method is called, all of the drivers are resumed, and user space is restarted.

If X is running, Linux switches from console to the GUI. In the case of IEGD and X, like the LeaveVT function, there is the analogous EnterVT (“enter Virtual Terminal”), which restores the previously saved graphics mode—the GUI display settings, saved register, and graphics device state information. After EnterVT restores everything fully, it saves the previous mode, i.e., the console mode before reentering the GUI.

With the kernel running again, all devices running, and user space applications, including X, running again the system has successfully resumed/woken-up from S3 Suspend to RAM.



What If Things Go Wrong?

Considering the number of components involved in power management, there are certainly many opportunities for failure. This section discusses common areas of consideration and resolution techniques. Although many of the items described are general to any Linux kernel, distribution, and devices in the system, the focus is on the graphics device.

System BIOS Issues

BIOS is responsible for platform initialization and configuration. BIOS issues can manifest themselves in a few ways. Some areas to consider are the following:

- Bugs in the APIC code prevent restoring a single register, resulting in some machines resuming without any interrupts being delivered.
- BIOS not setting a system register correctly or a register is being reset on resume to some default and not being set again either by BIOS or kernel or drivers. This can cause unpredictable results.
- Devices not being woken-up. On resume, power wells are not restored correctly making some devices inaccessible. In these cases, the device has power, but the device is no longer available to the system.

Video BIOS

Video BIOS (VBIOS) is a component of the platform's firmware, which also includes the System BIOS. Video BIOS is responsible for initializing and configuring the graphics device. Typically, the VBIOS is the first option ROM that executes in the platform firmware. This is essential, otherwise, no display would be available at boot. Some VBIOS issues related to S3 resume include the following:

- The ACPI specification does not require the BIOS to reprogram the video hardware so it may come back in an entirely unprogrammed state. Executing the code from VBIOS the same way the system BIOS does on machine startup is a possible workaround.
- Typically video repost is necessary to help Linux bring the console back after an S3; i.e., BIOS tells VBIOS to run its initialization code: video repost. After VBIOS re-initializes, the console is back. With IEGD, this will not work due to long standing and unchangeable architectural issues.
- In the case of VBIOS that supports video repost, e.g., GMA500 VBIOS, if it is enabled AND IEGD's IKM and X-Server are running, IEGD's Graphics Translation Tables (GTT) will be overrun. This results in incorrect mapping of GTT entries and a corrupted display. The workaround is to disable video repost in system BIOS.



- Cache values have been issues with power management. For greater performance, devices, such as IEGD tend to modify the default cache values within the system, i.e., cache values for regions or memory to achieve higher performance (see Wiki on [MTRR](#) for details). If the cache values are reset on resume and are not restored correctly, the graphics device performance may severely degrade, causing unusable, slow conditions, or visual side-effects, or both.

Miscellaneous Issues

Some general OS issues include:

- A driver's resume function may crash or cause a side-effect with another driver. Resume functions are called in series, just like when devices are suspended. A hang, crash, or any other issue can cause resource conflicts for other devices. Sometimes this situation happens because PCI resources can change at resume time.
- The Linux power management solution is inconsistent across various hardware platforms. As a result, the power management for some platform configurations work flawlessly and not for other configurations. This is an area where OS vendors or system integrators provide value-added services for integration and validation, including power management.
- Linux does not have a standard power management solution; however, ACPI is common, and, to a lesser extent, APM is present. These days HAL exists in some distros to support platform configuration, while others have their own solutions, such as acpid, GNOME DBUS/HAL, KDE KPowerSave, pm-utils for pm-suspend/pm-hibernate, SWUSP/TuxOnIce (S4), plus many user-developed power management scripts. This significantly complicates Linux power management.

Diagnosing PM Issues

It is often difficult to diagnose a power management issue because the display may be a blank. Below are some methods that can help determine the cause of power management problems.

- Disable the Intel graphics device. Enable SSH or VNC; i.e., boot headless. Try suspend and resume remotely. If suspend and resume work without issues, then there may be a graphics device issue, or a graphics driver issue, or both.
- Disable the graphics device and try an external graphics device, such as PCI or PCIE. Install the native driver and ensure that suspend and resume work. The objective is to isolate Intel graphics device and driver issues. If suspend/resume work fine, that may indicate an issue with the Intel graphics device issue, or a graphics driver issue, or both.



- Uninstall the IEGD driver. Use the non-embedded Intel GMA driver. Observe suspend/resume results. If suspend/resume work fine, this implies that the graphics device is working as it should, but there may be an IEGD driver issue.
- Recompile the kernel for specific CPU/hardware, setting the correct options for the processor. Take care to choose the right setting for the CPU that correctly implements power management/speed throttling features. At this stage, take this opportunity to fine-tune the kernel to choose the correct options for the devices with the platform.
- If the VBIOS supports it and a BIOS option is available, enable video repost. Try suspend and resume. Observe the state of the graphics device after resume.
- If you are using the IEGD driver, including both IKM and graphics driver, on an unsupported Linux distribution, ensure the IKM is correctly ported to the kernel. Please see the white paper, *IEGD Linux Kernel Module - Patching and Porting Methods* on <http://edc.intel.com/Software/Downloads/IEGD/#download> for guidance.
- Tweak `xorg.conf`. IEGD supports a large number of graphics options. Those options are configurable inside `xorg.conf`. The strategy: turn off any non-critical features and create an `xorg.conf` with minimal configuration for a single display with minimal features available. For debugging purposes, turn off twin/clone support, hardware acceleration or any other acceleration, and disable 2D acceleration and DRI (support for OpenGL), etc. Disabling various features used by the IEGD driver decreases the driver's exposure to more complex areas of the graphic hardware.
- Try using the VGA or VESA graphics driver. To do this, modify `xorg.conf` to use the standard VGA or VESA driver instead of "iegd" and then try suspend/resume. If it works, that may provide evidence of the platform's ability to successfully go into and out of a power managed state.
- The IEGD driver supports several port types, such as LVDS, SDVO, Analog, and HDMI. Because the IEGD driver uses different port drivers for each port type, it may be valuable to try a different port types. This can help identify port issues and potential display timing issues.
- The IEGD driver is highly customizable. The CED tool lets you configure IEGD using a number of options. One key configuration option is the ability to specify the display panel type and timings per the manufacturer's settings. Setting the timing correctly optimizes configuration for the display device. This helps ensure the display is set correctly at each stage of the power transitions.

The suggestions above may help diagnose and resolve power management issues. Frequently on S3 Resume, for example, the system attempts to resume but the screen will be blank, leading to the assumption that the graphics driver is at fault. Sometimes the graphics device or graphics driver is to blame, but not always. The hope is that by employing some of the techniques above, IEGD customers can better identify the cause of power management problems.



Conclusion

Implementing PM in Linux requires understanding the complex interaction among several hardware and software components in a given system. When a system resumes from a power state, such as S3, there is state transition among several variables (BIOS, VBIOS, kernel, drivers, windows manager, and X) to put the system back to its previous state.

When there is lack of display after resuming from S3 suspend state, the graphics driver may seem to be the cause. However, this behavior is usually the result of bugs in device drivers, BIOS, or VBIOS, or some combination.

Some common techniques can be used to discover the root cause of the blank display: booting headless, using an external graphics card, using open Linux graphics driver, etc. Some of the possible solutions involve disabling video repost and optimizing `xorg.conf`.

§

The Intel® Embedded Design Center provides qualified developers with web-based access to technical resources. Access Intel Confidential design materials, step-by step guidance, application reference solutions, training, Intel's tool loaner program, and connect with an e-help desk and the embedded community. Design Fast. Design Smart. Get started today. <http://intel.com/embedded/edc>.



Glossary

ACPI	Advanced Configuration and Power Interface. Besides being the power management specification, it provides the OS with a list of devices on the system. It also provides information about interrupt routing, e.g., if someone has just removed a hot-pluggable device from the system.
acpid	The user-space daemon needed to make the Linux ACPI support completely functional.
AGPGART	A kernel module for Linux that supports the extra data transfer features of Accelerated Graphics Port (AGP) video cards via the Graphics Address Remapping Table (GART).
APM	Advanced Power Management.
DSDT	<p>Differentiated System Description Table. The DSDT contains the Differentiated Definition Block, which supplies the implementation and configuration information about the base system. DSDT is provided by system OEM. The OS always inserts the DSDT information into the ACPI Namespace at system boot time and never removes it.</p> <p>The DSDT is in a bytecode called ACPI Machine Language (AML), compiled from a simple language called ACPI Source Language (ASL). At boot time, the system reads the DSDT, parses it, and executes various methods. These can do pretty much anything, but on the bright side they are being executed in kernel context and, in principle, you can filter out anything that you really do not want to do, such as overwriting CMOS.</p> <p>The final relevant piece of ACPI information is something called the FADT, or Fixed Address Descriptor Table. This gives the OS information about various register addresses. It's a static structure, and does not contain any executable code.</p>
Embedded Controller	The embedded controller performs complex low-level functions through a simple interface to the host microprocessor.
FADT	Fixed ACPI Description Table (ACPI).
GNOME	A desktop environment that runs on top of the Linux kernel.
GTT	Graphics Translation Table.
HAL	A Hardware Abstraction Layer (HAL) is implemented in software between the physical hardware of a computer and the software running on that computer.
IEGD	Intel® Embedded Graphics Drivers.



KDE A free software project based around a cross-platform desktop environment designed to run on Linux.

VBIOS Video BIOS.

References

Matthew Garrett blog, "How Linux suspend and resume works in the ACPI age," posted February 7, 2007. <http://www.advogato.org/article/913.html>

Vaddagiri, Srivatsa, Anand K Santhanam, Vijay Sukthankar and Murali Iyer. 2004. Power Management in Linux Based Systems *Linux Journal*, no. 119 (March 2004).

ACPI Spec 3.0b – acpi.info

Authors

Mudit Vats is a Graphics Software Engineer at Intel Corporation.

Ishu Verma is a Software Technical Marketing Engineer at Intel Corporation.



INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

This paper is for informational purposes only. THIS DOCUMENT IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. Intel disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein.

BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino logo, Core Inside, Dialogic, FlashFile, i960, InstantIP, Intel, Intel logo, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Intel Core, Intel Inside, Intel Inside logo, Intel. Leap ahead., Intel. Leap ahead. logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel EP80579 Integrated Processor, Intel StrataFlash, Intel Viiv, Intel vPro, Intel XScale, IPLink, Itanium, Itanium Inside, MCS, MMX, Oplus, OverDrive, PDCharm, Pentium, Pentium Inside, skool, Sound Mark, The Journey Inside, VTune, Xeon, and Xeon Inside are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2010 Intel Corporation. All rights reserved.