

**IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
WACO DIVISION**

LIBERTY PATENTS, LLC,

Plaintiff,

v.

DELL TECHNOLOGIES INC. and
DELL INC.,

Defendants.

CIVIL ACTION NO. 6:20-cv-944

ORIGINAL COMPLAINT FOR
PATENT INFRINGEMENT

JURY TRIAL DEMANDED

ORIGINAL COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Liberty Patents, LLC (“Liberty Patents” or “Plaintiff”) files this original complaint against Defendants Dell Technologies, Inc. and Dell Inc. (collectively “Dell” or “Defendants”), alleging, based on its own knowledge as to itself and its own actions and based on information and belief as to all other matters, as follows:

PARTIES

1. Liberty Patents is a limited liability company formed under the laws of the State of Texas, with its principal place of business at 2325 Oak Alley, Tyler, Texas 75703.
2. Defendant Dell Technologies Inc. is a corporation organized and existing under the laws of Delaware. Dell Technologies Inc. may be served with process through its registered agent, Corporation Service Company located at 251 Little Falls Drive, Wilmington, Delaware 19808.
3. Defendant Dell Inc. is a corporation organized and existing under the laws of Delaware. Dell Inc. may be served with process through its registered agent, Corporation Service Company d/b/a/ CSC-Lawyers Incorporating Service Company, 211 E. 7th Street, Suite 620, Austin, Texas, 78701. Dell Inc. is an indirect subsidiary of Dell Technologies Inc.

JURISDICTION AND VENUE

4. This is an action for infringement of a United States patent arising under 35 U.S.C. §§ 271, 281, and 284–85, among others. This Court has subject matter jurisdiction of the action under 28 U.S.C. § 1331 and § 1338(a).

5. This Court has personal jurisdiction over the Dell Defendants pursuant to due process and/or the Texas Long Arm Statute because, *inter alia*, (i) Dell has done and continues to do business in Texas; (ii) Dell has committed and continues to commit acts of patent infringement in the State of Texas, including making, using, offering to sell, and/or selling accused products in Texas, and/or importing accused products into Texas, including by Internet sales and sales via retail and wholesale stores, inducing others to commit acts of patent infringement in Texas, and/or committing a least a portion of any other infringements alleged herein, and (iii) Dell Inc. is registered to do business in Texas.

6. Venue is proper in this district as to Defendants pursuant to 28 U.S.C. § 1400(b). Venue is further proper because Dell has committed and continue to commit acts of patent infringement in this district, including making, using, offering to sell, and/or selling accused products in this district, and/or importing accused products into this district, including by Internet sales and sales via retail and wholesale stores, inducing others to commit acts of patent infringement in this district, and/or committing at least a portion of any other infringements alleged herein in this district.

7. Dell has a regular and established place of business in this district, including at One Dell Way, Round Rock, Texas 78682. Depicted below are additional Dell locations within this district:



Source: <https://goo.gl/maps/3TTbbFFptgNseq5V6> (401 Dell Way)



Source: <https://goo.gl/maps/S8KKb5JoaKP6cEx76> (200 Dell Way)

BACKGROUND

8. The two patents-in-suit cover technology used in computer systems, such as notebook computers, laptop computers, desktop computers, tablets, and other electronic devices. More particularly, the patents-in-suit describe key improvements to electronic devices in the

areas of better power distribution and power management, and a better process for retrieving automatic software updates.

9. U.S. Patent No. 6,920,573 (“the ’573 Patent”) generally relates to a system for conserving energy in electronic systems. Specifically, the inventor developed a system that provides much-needed energy savings for computers, such as notebooks and laptops, by including various operating modes that limit power usage. In particular, the ’573 Patent describes three operating modes. The first mode is a regular operating mode where the electronic device is fully powered on and where the main microprocessor is running. The second mode is a power-saving mode where the main microprocessor is not running, yet the system is still activated. The third mode is also a power-saving mode, and more specifically, a standby mode from which the first mode can be activated. The ’573 Patent also discloses components to power the system, such as a rechargeable battery, and components to control the system, such as a power button.

10. Major companies in the electronics industry have cited the invention of the ’573 Patent during patent prosecution. Indeed, the ’573 Patent has been cited over fifty times by leading companies, including Broadcom, Compal Electronics, Foxconn, Google, Hewlett-Packard, Intel, Panasonic (Matsushita), Microsoft, NVIDIA, Sony, Toshiba, Transmeta, and Wistron.

11. U.S. Patent No. 7,493,612 (“the ’612 Patent”) discloses systems and methods for automatically updating the system software of an embedded system. Claim 1 describes an embedded system capable of automatically updating system software using update agent interface programming (UAIP)—code that initiates an update of the system software during the boot process. The embedded system includes first system software and a boot image. The

system also includes a micro-controller capable of transforming the first system software into system code and the boot image into boot code. The boot code includes update agent interface programming (UAIP) for initiating updating of the first system software before executing the system code. The system can be coupled to an external data storage device, which contains the second system software (i.e., the updated system code). If there is an update to the system software, the second system software is read from the external data storage device. As a result of the '612 Patent's inventive system, a computer can advantageously retrieve automatic updates during boot without loading its outdated OS—a more efficient, time-saving solution.

12. The '612 Patent's inventive system was developed by the Taiwanese company, Lite-On Technology Corp., which develops a wide range of consumer electronics products, such as semiconductors, monitors, motherboards, etc. Lite-On was originally founded in 1975 by former employees of Texas Instruments. While the company originally developed LEDs, it branched into other industries, such as embedded systems and related software, and stayed on the forefront of developing technologies. Lite-On was recently purchased by the Japanese company, Kioxia—a former division of Toshiba—for \$165 million.

13. The '612 Patent discloses a novel and important invention that is highly relevant to today's technology, which relies heavily on recurring updates to computer systems and IoT devices. It has been cited by major technology companies like Foxconn, Google, Hewlett-Packard, IBM, Samsung, Silicon Graphics, and Texas Instruments.

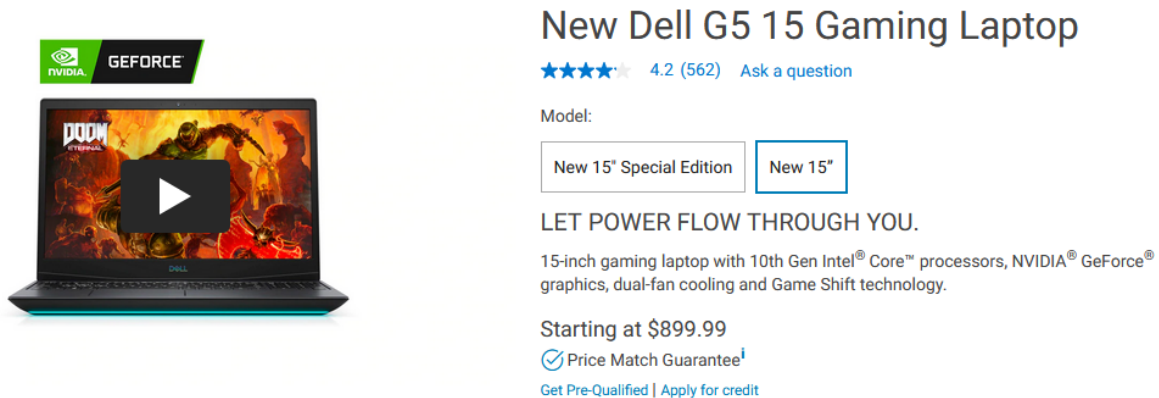
COUNT I

DIRECT INFRINGEMENT OF U.S. PATENT NO. 6,920,573

14. On July 19, 2005, the '573 Patent was duly and legally issued by the United States Patent and Trademark Office for an invention entitled “Energy-Conserving Apparatus and Operating System Having Multiple Operating Functions Stored in Keep-Alive Memory.”

15. Liberty Patents is the owner of the '573 Patent, with all substantive rights in and to that patent, including the sole and exclusive right to prosecute this action and enforce the '573 Patent against infringers, and to collect damages for all relevant times.

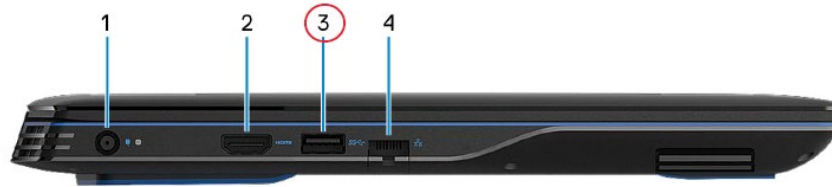
16. Dell made, had made, used, imported, provided, supplied, distributed, sold, and/or offered for sale products and/or systems including, for example, its Dell G5 15 Laptop and other products including the “PowerShare” feature¹ (“accused products”):

A screenshot of the Dell G5 15 Gaming Laptop product page. On the left is a black laptop with a red and orange 'DOOM Eternal' game cover on the screen. Above the laptop is a green NVIDIA GeForce logo. To the right of the laptop, the text reads: 'New Dell G5 15 Gaming Laptop', followed by a 4.2 star rating (562 reviews) and a link to 'Ask a question'. Below this is a 'Model:' section with two buttons: 'New 15" Special Edition' and 'New 15"'. The main headline is 'LET POWER FLOW THROUGH YOU.' followed by a description: '15-inch gaming laptop with 10th Gen Intel® Core™ processors, NVIDIA® GeForce® graphics, dual-fan cooling and Game Shift technology.' The price is listed as 'Starting at \$899.99' with a 'Price Match Guarantee' icon and links for 'Get Pre-Qualified' and 'Apply for credit'.

Source: <https://www.dell.com/en-us/shop/cty/pdp/spd/g-series-15-5500-laptop>

¹ See, e.g., Dell G3 15 (3500), G5 (5590), G7 15 (7500), Inspiron 15-7567, Inspiron 17 (5000), Inspiron 17 (7000), Inspiron i3158-3275SLV, Latitude 3390 2-in-1, Latitude 5400, Latitude 5420, Latitude 7424, Latitude 7480, Latitude E5470, Latitude E5550, Latitude E6230, Latitude E6430, Latitude E7470, Precision 5520, Vostro Business Laptop (876slv), XPS 13, XPS 13 (7390), XPS 13 (9343), XPS 15.

Computers shipped with NVIDIA GeForce GTX 1650



3. **USB 3.2 Gen 1 port with PowerShare**

Connect devices such as external storage devices and printers.

Provides data transfer speeds up to 5 Gbps. PowerShare enables you to charge your USB devices even when your computer is turned off.

Source: https://topics-cdn.dell.com/pdf/g-series-15-5500-laptop_setup-guide_en-us.pdf

17. By doing so, Dell has directly infringed (literally and/or under the doctrine of equivalents) at least Claim 13 of the '573 Patent. Dell's infringement of the '573 Patent is ongoing.

18. The Dell G5 15 Laptop is an information-processing apparatus with multiple operating functions. It includes a first group of circuitry that is actuatable to provide a first operating function. The first group of circuitry comprises main microprocessor circuitry.

19. For example, the Dell G5 15 Laptop includes a processor for performing various processing functions. The processor includes Arithmetic Logic Units (ALU), Instruction and Data Caches, and other blocks. The processor also has different states like working state, sleeping state, and off state etc., which correspond to the laptop's Power On mode, Sleep mode and Shut Down mode, respectively. The processor functions differently depending on the current operating mode.

20. During Power On mode, the processor provides processing functions, including application processing, graphics processing, etc. ("first operating function"). The processing blocks like ALU, FPU, memory, etc. ("first group of circuitry") consume power and implement

these required functions. These blocks are part of the core or Central Processing Unit (“main microprocessor circuitry”) of the processor.

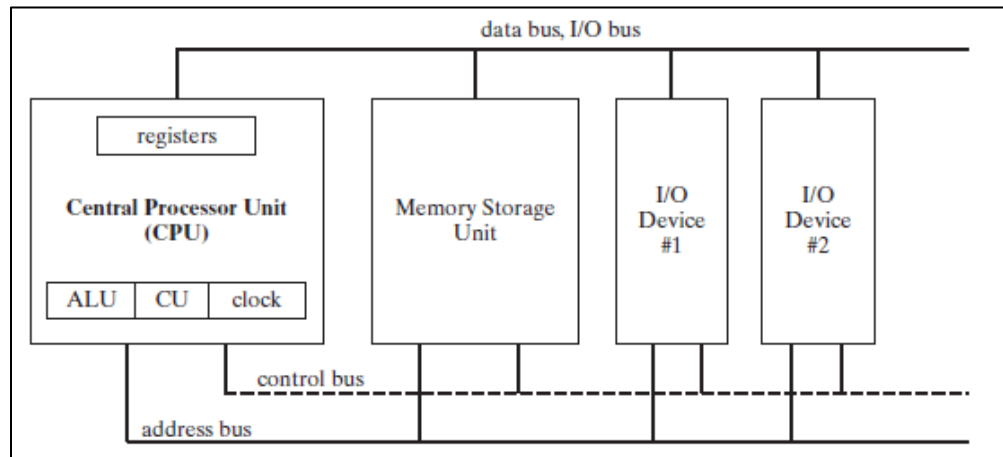
Immersive graphics: With discrete graphics options of up to NVIDIA® GeForce® GTX 1660Ti or RTX 2070 with Max-Q technology, you can game at higher settings with crystal-clear detail and smooth gameplay. Enjoy a powerful core streaming multiprocessor driving new Turing shading advancements and step up to RTX for powerhouse performance, next-gen graphics, and NVIDIA technologies like ray tracing, DLSS, and AI-enhanced graphics rendering.

Memory that lasts: Experience the action in all its glory with faster loading times and a quieter system powered by up to 16GB GDDR6 of dedicated memory.

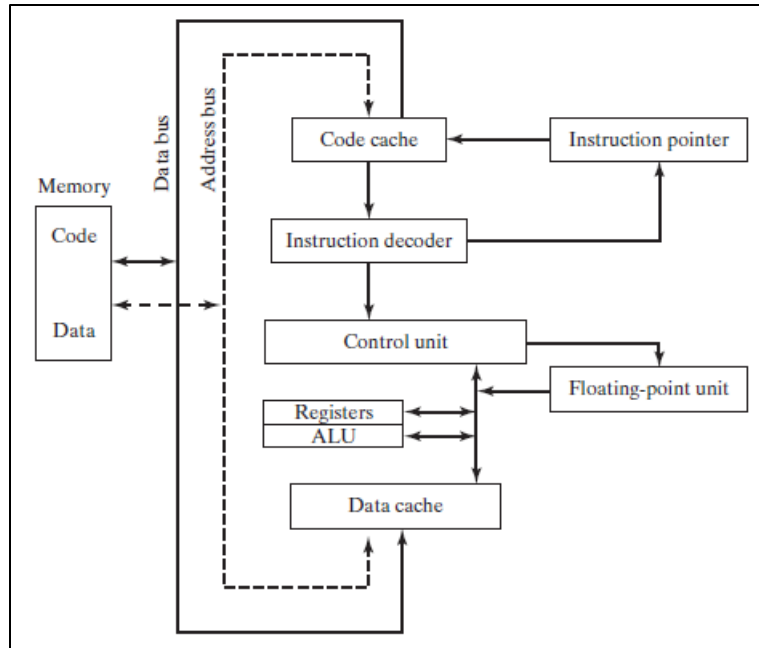
Get drawn in: Every detail of your game just got magnified thanks to an FHD display with two-sided narrow bezel and optional 144Hz panel.

Full speed ahead: Killer Gigabit Ethernet ensures blazing-fast speed and minimizes lag for smooth network traffic.

Source: <https://www.dell.com/en-us/shop/cty/pdp/spd/g-series-15-5500-laptop>



Source: <https://www.allaboutcircuits.com/technical-articles/an-introduction-to-x86-processor-architecture/>



Source: <https://www.allaboutcircuits.com/technical-articles/an-introduction-to-x86-processor-architecture/>

21. The following citations disclose different operating modes of the laptop, including Shut Down mode, Sleep mode, and Power On mode. The computer operates differently according to the current operating mode.



4. Power button with optional fingerprint reader

Press to turn on the computer if it is turned off, in sleep state, or in hibernate state.

When the computer is turned on, press the power button to put the computer into sleep state; press and hold the power button for 4 seconds to force shut-down the computer.

If the power button has a fingerprint reader, place your finger on the power button to log in.

i **NOTE:** You can customize power-button behavior in Windows. For more information, see *Me and My Dell* at www.dell.com/support/manuals.

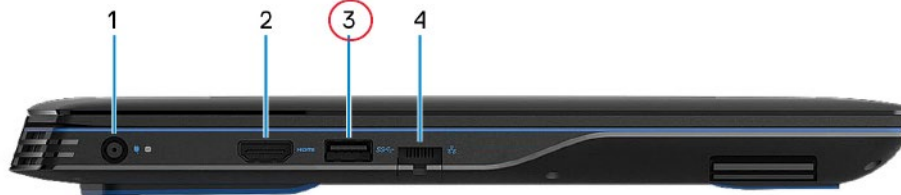
Source: https://topics-cdn.dell.com/pdf/g-series-15-5500-laptop_setup-guide_en-us.pdf (page 9)

22. The Dell G5 15 Laptop includes a second group of circuitry that is actuatable to provide a second operating function. During the second operating function, the system is not required to activate the main microprocessor circuitry.

23. For example, the Dell G5 15 Laptop has a “PowerShare” feature that allows a user to charge (“second operating function”) USB connected devices (such as such as a mobile devices, cameras, activity trackers, smartwatches, etc.) even when the laptop is in the Shut Down or Off mode. Mobile devices can be charged using the designated USB port having the “PowerShare” feature without requiring the laptop to be in a working state (i.e., Power On mode). The corresponding USB charger IC/USB board circuit (“second group of circuitry”) can be actuated to provide the charging function during Shut Down mode.

Left

Computers shipped with NVIDIA GeForce GTX 1650



3. USB 3.2 Gen 1 port with PowerShare

Connect devices such as external storage devices and printers.

Provides data transfer speeds up to 5 Gbps. PowerShare enables you to charge your USB devices even when your computer is turned off.

NOTE: If the charge on your computer's battery is less than 10 percent, you must connect the power adapter to charge your computer, and USB devices connected to the PowerShare port.

NOTE: If a USB device is connected to the PowerShare port before the computer is turned off or in hibernate state, you must disconnect and connect it again to enable charging.

NOTE: Certain USB devices may not charge when the computer is turned off or in sleep state. In such cases, turn on the computer to charge the device.

Source: https://topics-cdn.dell.com/pdf/g-series-15-5500-laptop_setup-guide_en-us.pdf (page 7)

24. The Dell G5 15 Laptop includes a third group of circuitry that is actuatable to provide a standby function that allows the first group of circuitry (when deactivated) to be reactuatable so that it can provide the first operating function. The third group of circuitry also comprises keep-alive memory circuitry for storing information needed for resuming the first operating function or the second operating function.

25. For example, the Dell G5 15 Laptop includes different operating modes like Sleep mode, Power On mode, and Shut Down mode. The Sleep mode (“standby function”) can be activated and deactivated (i.e., to wake up the system) by pressing the Power button. The laptop includes corresponding circuitry (“third group of circuitry”) that activates and deactivates the Sleep mode.

26. During Sleep mode, computational tasks are not performed, and the system consumes less power. The system retains enough context in order to return to a working state (“resuming said first operating function”) by storing or saving information in hardware memory, such as RAM or in a disk (“keep-alive memory circuitry”).



4. Power button with optional fingerprint reader

Press to turn on the computer if it is turned off, in sleep state, or in hibernate state.

When the computer is turned on, press the power button to put the computer into sleep state; press and hold the power button for 4 seconds to force shut-down the computer.

If the power button has a fingerprint reader, place your finger on the power button to log in.

NOTE: You can customize power-button behavior in Windows. For more information, see *Me and My Dell* at www.dell.com/support/manuals.

Source: https://topics-cdn.dell.com/pdf/g-series-15-5500-laptop_setup-guide_en-us.pdf (page 9)

- S1/S2/S3 - **Sleep:** Your PC appears to be off. It uses less power than leaving the system powered on but uses more power than hibernation. S3 consumes less power than S2, and S2 consumes less power than S1. Your system will usually only support one of these three states. In these states, the PC's volatile memory is kept refreshed to maintain the system state when it went to sleep. Some internal components continue to get power so the computer can wake from input from the keyboard, LAN, or a USB device. (This is much like pausing and unpausing the system.)

Source: <https://www.dell.com/support/article/en-us/sln309800/windows-10-troubleshooting-sleep-hibernation-issues-on-your-dell-pc?lang=en>

27. The Dell G5 15 Laptop includes power providing means for providing power to the first group of circuitry, the second group of circuitry, and the third group of circuitry.

28. For example, the Dell G5 15 Laptop includes a battery (“power providing means”) for providing power to the different circuits present in the system, including the CPU, memory, and I/O Peripherals (which include USB).

The following table lists the power adapter specifications of your Dell G5 15 5500.

Table 17. Power adapter specifications

Description	Option one	Option two
Type	E4 180W	E4 240W

The following table lists the battery specifications of your Dell G5 15 5500.

Table 18. Battery specifications

Description	Option one	Option two
Battery type	3-cell polymer (51 Wh)	4-cell polymer (68 Wh)

Source: https://topics-cdn.dell.com/pdf/g-series-15-5500-laptop_setup-guide_en-us.pdf (page 19)

29. The Dell G5 15 Laptop includes control means for controlling said power providing means to selectively activate said first group of circuitry, said second group of circuitry, and said third group of circuitry, so as to respectively provide said first operating function, said second operating function, and said standby function.

30. For example, the Dell G5 15 Laptop includes different operating modes like Power On, Sleep, and Shut Down modes. Sleep mode (“standby function”), Shut Down mode, and Power On mode (which provides “first operating function”) can be activated using the Power button (“control means”). The USB port with the PowerShare feature enables charging of a mobile device through the designated USB port during Shut Down mode (“second operating function”).

31. The processor of the Dell G5 15 Laptop includes a Power Management Integrated Circuit (PMIC) that manages the power distribution in the processor system. The PMIC provides power to different circuits of the processor system. Further, the PMIC receives control

inputs from the processor system, i.e., signals from the power button are used by PMIC as control inputs for enabling and disabling the power distribution for the circuits in the processor system.



4. Power button with optional fingerprint reader

Press to turn on the computer if it is turned off, in sleep state, or in hibernate state.

When the computer is turned on, press the power button to put the computer into sleep state; press and hold the power button for 4 seconds to force shut-down the computer.

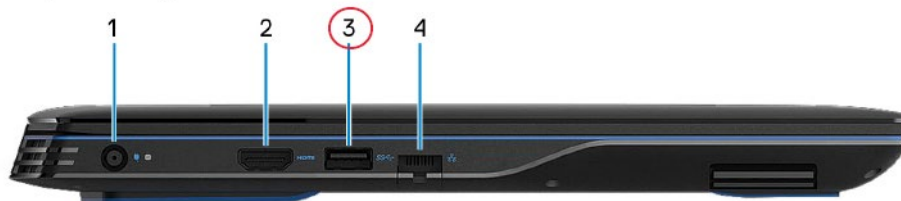
If the power button has a fingerprint reader, place your finger on the power button to log in.

NOTE: You can customize power-button behavior in Windows. For more information, see *Me and My Dell* at www.dell.com/support/manuals.

Source: https://topics-cdn.dell.com/pdf/g-series-15-5500-laptop_setup-guide_en-us.pdf (page 9)

Left

Computers shipped with NVIDIA GeForce GTX 1650



3. USB 3.2 Gen 1 port with PowerShare

Connect devices such as external storage devices and printers.

Provides data transfer speeds up to 5 Gbps. PowerShare enables you to charge your USB devices even when your computer is turned off.

NOTE: If the charge on your computer's battery is less than 10 percent, you must connect the power adapter to charge your computer, and USB devices connected to the PowerShare port.

NOTE: If a USB device is connected to the PowerShare port before the computer is turned off or in hibernate state, you must disconnect and connect it again to enable charging.

NOTE: Certain USB devices may not charge when the computer is turned off or in sleep state. In such cases, turn on the computer to charge the device.

Source: https://topics-cdn.dell.com/pdf/g-series-15-5500-laptop_setup-guide_en-us.pdf (page 7)

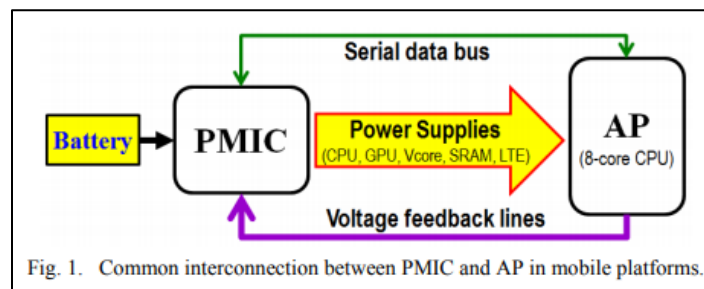


Fig. 1. Common interconnection between PMIC and AP in mobile platforms.

Source: <https://ieeexplore.ieee.org/document/7237388>

32. Dell has had knowledge of the '573 Patent at least as of the date when it was notified of the filing of this action.

33. Liberty Patents has been damaged as a result of Dell's infringing conduct alleged above. Thus, Dell is liable to Liberty Patents in an amount that adequately compensates it for such infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

34. Liberty Patents and/or its predecessors-in-interest have satisfied all statutory obligations required to collect pre-filing damages for the full period allowed by law for infringement of the '573 Patent.

COUNT II

DIRECT INFRINGEMENT OF U.S. PATENT NO. 7,493,612

35. On February 17, 2009, the '612 Patent was duly and legally issued by the United States Patent and Trademark Office for an invention entitled “Embedded System and Related Method Capable of Automatically Updating System Software.”

36. Liberty Patents is the owner of the '612 Patent, with all substantive rights in and to that patent, including the sole and exclusive right to prosecute this action and enforce the '612 Patent against infringers, and to collect damages for all relevant times.

37. Dell made, had made, used, imported, provided, supplied, distributed, sold, and/or offered for sale products and/or systems including, for example, its family of products running Chrome OS,² including the Dell Latitude 5300 2-in-1 Chromebook Enterprise (“accused products”):



Latitude 5300 2-in-1 Chromebook Enterprise

Model:

13" 2-in-1 Chromebook Enterprise	14" Chromebook Enterprise	New 14" 7410
----------------------------------	---------------------------	--------------

Latitude and Chrome Enterprise. Smarter together.

World's most powerful 13-inch Chromebook Enterprise 2-in-1.* Featuring Dell ProSupport, Chrome Enterprise and latest 8th Gen Intel® processors.

[Choose Tech Specs](#)

² See, e.g., Dell Chromebook 11, Chromebook 11 (3120), Chromebook 11 (3180), Chromebook 11 (5190), Chromebook 11 2-in-1 (3189), Chromebook 11 2-in-1 (5190), Chromebook 3100 2-in-1, Chromebook 13 (7310), Chromebook 13 (3380), Chromebook 3100, Chromebook 3400, Inspiron Chromebook 14 2-in-1 (7486), Latitude 5300 2-in-1 Chromebook Enterprise Latitude 5400 Chromebook Enterprise, Latitude 7410 Chromebook Enterprise, Latitude 7410 2-in-1 Chromebook Enterprise, Chromebox, etc.

Source: <https://www.dell.com/en-us/work/shop/laptops/13-2-in-1-chromebook-enterprise/spd/latitude-13-5300-2-in-1-chrome-laptop>

Operating System - Google Chrome OS



Source: <https://www.dell.com/en-us/work/shop/laptops/13-2-in-1-chromebook-enterprise/spd/latitude-13-5300-2-in-1-chrome-laptop>

Auto Update policy

Overview

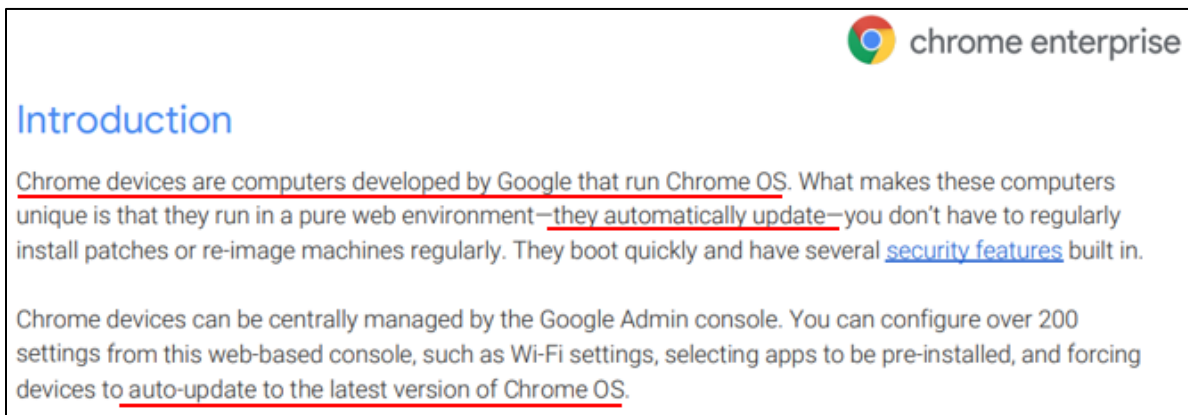
Chrome devices (e.g. Chromebook, Chromebox, Chromebase, Chromebit) receive automatic updates that enhance both the device and its software. Device updates provide the latest features and keep the device secure, and are applied across the operating system, browser and hardware. These updates depend on many device specific non-Google hardware and software providers that work with Google to provide the highest level of security and stability support. For this reason, older Chrome devices cannot receive updates indefinitely to enable new OS and browser features.

Source: <https://support.google.com/chrome/a/answer/6220366?hl=en>

38. By doing so, Dell has directly infringed (literally and/or under the doctrine of equivalents) at least Claim 1 of the '612 Patent. Dell's infringement in this regard is ongoing.

39. The Dell Latitude 5300 2-in-1 Chromebook Enterprise includes an embedded system that is capable of automatically updating system software.

40. For example, the Dell Latitude 5300 2-in-1 Chromebook Enterprise runs Chrome OS, which can automatically update the device's firmware over the internet. Chromebooks include an Embedded Controller (EC), which is responsible for power sequencing of the main CPU or the Application Processor (AP), keyboard control, thermal control, battery charging control, verified boot, etc. The EC—together with the components it controls—is an embedded system capable of automatically updating the system software. Specifically, the system includes two types of firmware: RO firmware and RW firmware. The RW firmware (“system software”) is stored in the updateable section of the firmware and can be automatically updated.



Source: https://services.google.com/fh/files/misc/chrome_device_deployment_guide.pdf

Auto Update policy

Overview

Chrome devices (e.g. Chromebook, Chromebox, Chromebase, Chromebit) receive automatic updates that enhance both the device and its software. Device updates provide the latest features and keep the device secure, and are applied across the operating system, browser and hardware. These updates depend on many device specific non-Google hardware and software providers that work with Google to provide the highest level of security and stability support. For this reason, older Chrome devices cannot receive updates indefinitely to enable new OS and browser features.

Source: <https://support.google.com/chrome/a/answer/6220366?hl=en>

ChromeOS Firmware Updater

This repository contains the firmware updater (`chromeos-firmwareupdate`) that will update firmware images related to verified boot, usually host (also known as AP, BIOS or MAIN) and EC (Embedded Controller).

Contents

- Introduction
- Using Firmware Updater
 - Update manually
 - Simulating ChromeOS Auto Update
- Building Firmware Updater
- Manipulating Firmware Updater Packages
 - CROS_FIRMWARE_MAIN_IMAGE
 - CROS_FIRMWARE_MAIN_RW_IMAGE
 - CROS_FIRMWARE_EC_IMAGE
- Technical Details
 - Packaging
 - Updater logic

Introduction

Auto update is one of the most important feature in Chrome OS. Updating firmware is one of the most complicated process, since all Chromebooks come with firmware that implemented verified boot and must be able to update in background silently.

Source:

<https://chromium.googlesource.com/chromiumos/platform/firmware/+master/README.md>

What's an Embedded Controller anyway?

- A tiny SoC that manages battery charging, fans, keyboard, LEDs, etc.
- Typically runs even when the main system processor is off
 - We call the main system CPU the "AP" (for Application Processor)
- Most laptops have them
- Most Chromebooks do too
- Ours is open source, which is unusual

Source: https://www.coreboot.org/images/5/50/An_Open_Source_EC.pdf (Page 4).

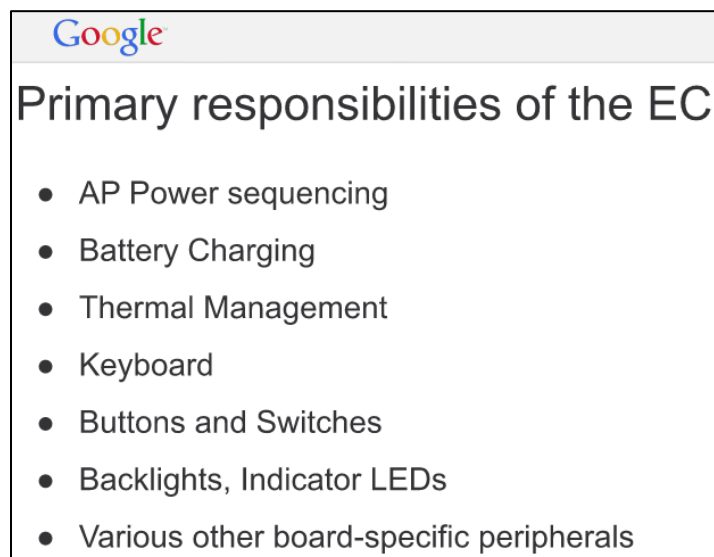
Introduction

The Chromium OS project includes open source software for embedded controllers (EC) used in recent ARM and x86 based Chromebooks. This software includes a lightweight, multitasking OS with modules for power sequencing, keyboard control, thermal control, battery charging, and verified boot. The EC software is written in C and supports a variety of micro-controllers.

This document is a guide to help make you familiar with the EC code, current features, and the process for submitting code patches.

For more see the Chrome OS Embedded Controller [presentation](#) and [video](#) from the 2014 Firmware Summit.

Source: <https://chromium.googlesource.com/chromiumos/platform/ec/>



Google

Primary responsibilities of the EC

- AP Power sequencing
- Battery Charging
- Thermal Management
- Keyboard
- Buttons and Switches
- Backlights, Indicator LEDs
- Various other board-specific peripherals

Source: https://docs.google.com/presentation/d/1Xa_Z5SjW-soPvkugAR8_TEJFrJpzoZUa9HNR14Sjs8/pub?start=false&loop=false&delayms=3000&slide=id.g2bc16935c_0142 (Slide 20).

Chrome EC

- Embedded Controllers are vital but closed
- Chrome EC is open source
 - chromiumos/platform/ec.git
- Chrome EC is designed for security
 - RO and RW regions
 - RW update is signed and handled by host firmware
 - EC Software Sync is part of Verified Boot
- Support for different ARM SOCs
 - Texas Instruments Stellaris Cortex-M4
 - ST Micro STM32 Cortex-M3
 - More in progress...

Source: https://docs.google.com/presentation/d/1h-nsDGIQmYI21dr95nYgLmyCYDgBIpJWSt9b7AqTZaw/pub?start=false&loop=false&delayms=3000&slide=id.g2b77a1dcf_298 (Slide 29).

Citation 7:

Chrome EC Software Sync

It is important that the AP firmware (coreboot) and the EC firmware remain compatible through upgrades. During every Normal Mode boot, the EC firmware is verified by the AP firmware and updated, if required. In Recovery Mode, the EC and AP firmware stay in read-only mode.

Source: <https://link.springer.com/content/pdf/10.1007%2F978-1-4842-0070-4.pdf> (Page 119).

Terminology

RO and RW

MCUs running the EC code have read-only (RO) and read-write (RW) firmware. Coming out of reset, the MCU boots into its RO firmware.

In the case of the EC, the RO firmware boots the host and asks it verify a hash of the RW firmware (software sync). If the RW firmware is invalid, it is updated from a copy in the host's RW firmware.

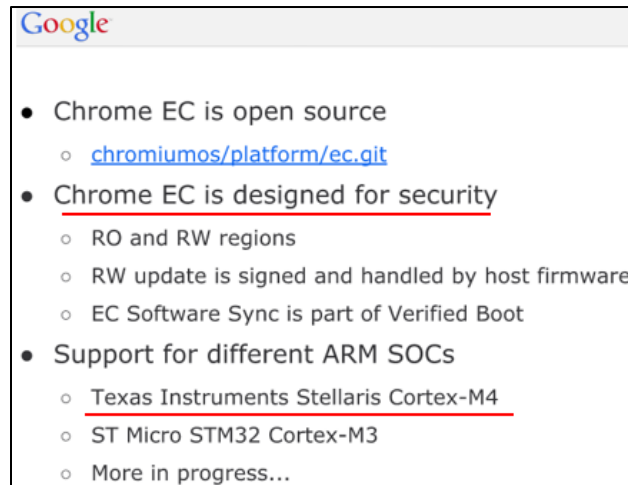
Source:

https://chromium.googlesource.com/chromiumos/platform/ec/+master/docs/write_protection.md

A feature called "Software Sync" keeps a copy of the read-write (RW) EC firmware in the RW part of the system firmware image. At boot, if the RW EC firmware doesn't match the copy in the system firmware, the EC's RW section is re-flashed.

Source: <https://chromium.googlesource.com/chromiumos/platform/ec/>

41. The Dell Latitude 5300 2-in-1 Chromebook Enterprise includes a first storage device for storing a first system software and a boot image. For example, the device includes flash memory ("first storage device") that stores the RO firmware ("boot image") and RW firmware ("first system software").



Source: https://docs.google.com/presentation/d/1Xa_Z5SjW-soPvkugAR8_TEJFrJpzoZUa9HNR14Sjs8/pub?start=false&loop=false&delayms=3000&slide=id.g2bbed09ac_111 (Slide 2).

Changing Software Write Protection with `ectool``ectool flashprotect`

Print out current flash protection state.

```
Flash protect flags: 0x0000000f wp_gpio_asserted ro_at_boot ro_now all_now
Valid flags:        0x0000003f wp_gpio_asserted ro_at_boot ro_now all_now STUCK INCONSISTENT
Writable flags:     0x00000000
```

`Flash protect flags` - Current flags that are set.

`Valid flags` - All the options for flash protection.

`Writable flags` - The flags that currently can be changed. (In this case, no flags can be changed).

Flags:

- `wp_gpio_asserted` - Whether the hardware write protect GPIO is currently asserted (read only).
- `ro_at_boot` - Whether the EC will write protect the RO firmware on the next boot of the EC.
- `ro_now` - Protect the read-only portion of flash immediately. Requires hardware WP be enabled.
- `all_now` - Protect the entire flash (including RW) immediately. Requires hardware WP be enabled.
- `STUCK` - Flash protection settings have been fused and can't be cleared (should not happen during normal operation. Read only.)
- `INCONSISTENT` - One or more banks of flash is not protected when it should be (should not happen during normal operation. Read only.).

Source:

https://chromium.googlesource.com/chromiumos/platform/ec/+/master/docs/write_protection.md

Firmware Image

The Chrome OS firmware image has two main sections: Read-Only (RO) and Read-Write (RW). The RO firmware is set at the factory and cannot be updated after manufacturing. The RW firmware can be updated during Chrome OS auto-update (AU).

If a problem is found in RO firmware, Google creates an update and places it in the RW firmware. During the boot process, the RO firmware checks whether there is an update in the RW section and, if so, jumps to the RW update to execute the new boot code.

The RO firmware contains the following code:

- U-Boot, including the device tree for this system
- On x86 systems: coreboot
- Google Binary Block (GBB), which contains the following:
 - Recovery screen images
 - Public keys needed to verify the RW firmware
- Firmware ID (a string with the version number and device type)

The RW firmware contains two sections: A and B. Each section contains the following:

- U-Boot, including the device tree for this system (identical to the U-Boot images in RO firmware)
- VBlock, which contains the signatures used to verify the kernel before loading and running it
- Firmware ID
- Embedded Controller image
- Fmap, a data structure that describes the layout and contents of the SPI Flash. This structure is required by the Flashrom tool.

Source: [https://www.chromium.org/chromium-os/firmware-porting-guide/2-](https://www.chromium.org/chromium-os/firmware-porting-guide/2-concepts?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1)

[concepts?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1](https://www.chromium.org/chromium-os/firmware-porting-guide/2-concepts?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1)

42. The RO firmware and RW firmware are stored in system memory.

BASE	SIZE	SECTION	DESCRIPTION
0x000000	0x200000	SI_ALL	Descriptor + ME
0x200000	0x0f0000	RW_SECTION_A	Read-Write Firmware A
0x2f0000	0x0f0000	RW_SECTION_B	Read-Write Firmware B
0x3e0000	0x010000	RW_MRC_CACHE	Memory Training Cache
0x3f0000	0x004000	RW_ELOG	Event Log
0x3f4000	0x004000	RW_SHARED	Shared Data
0x3f8000	0x002000	RW_VPD	Read-Write VPD
0x400000	0x200000	RW_LEGACY	Legacy Firmware
0x600000	0x004000	RO_VPD	Read-Only VPD
0x610000	0x000800	FMAP	Flash Map
0x610800	0x000040	RO_FRID	RO Firmware ID
0x611000	0x0ef000	GBB	Google Binary Block
0x700000	0x100000	BOOT_STUB	Read-Only Firmware

Source: https://docs.google.com/presentation/d/1h-nsDGIQmYI21dr95nYgLMYCYDgBIpJWSt9b7AqTZaw/pub?start=false&loop=false&delayms=3000&slide=id.g2b77a1dcf_1128 (Slide 33).

43. The Dell Latitude 5300 2-in-1 Chromebook Enterprise includes a micro-controller that is coupled to the first storage device for respectively transforming the first system software and the boot image into system code and boot code. The micro-controller orderly executes the boot code and the system code to control booting of the embedded system.

44. For example, the EC (“micro-controller”) is connected to flash memory (“first storage device”) that stores both the RO firmware (“boot image”) and RW firmware (“first system software”). The firmware can have different configurations, which relates to how it uses flash and RAM memory for system code and boot code execution. For example, firmware images in Chrome OS support configurations like Internal Mapped Storage, Code Copied to RAM for Use, etc.

45. During system boot, linker scripts provide information as to how different sections of the RO firmware (“boot image”) and RW firmware (“first system software”) map to different sections of memory. The system provides linker scripts for both the RO firmware and the RW firmware in all the supported memory configurations. Before booting, a small program called Boot Loader (i.e., start-up code) uses linker scripts to load sections of the RO firmware and RW firmware into different parts of memory. The EC executes the mapped RO firmware (“boot code”) and the mapped RW firmware (“system code”) only after they are loaded into specified sections of memory determined by linker scripts. Accordingly, the RO firmware (“boot image”) and RW firmware (“first system software”) are transformed into memory-mapped executable code (i.e., “boot code” and “first system code”) after being mapped and loaded into their respective sections of memory.

EC Image Geometry Spec

Introduction

The EC codebase currently supports the following chips:

lm4, stm32: Internal memory-mapped flash storage

cr50: Internal memory-mapped flash storage, with image signature preceding RO image

npcx: External memory-mapped flash storage dedicated for EC, code copied to SRAM before execution

mec1322: External unmapped flash storage dedicated for EC, code is copied to SRAM before execution

For memory-mapped flash storage, contents are read from a chip-defined region in memory. Contents are written through SPI commands.

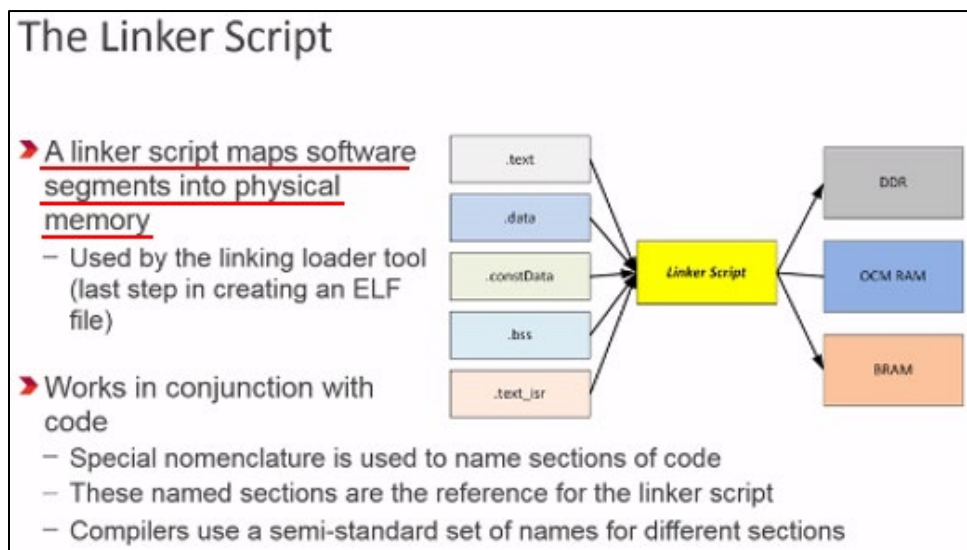
Source: <https://www.chromium.org/chromium-os/ec-development/ec-image-geometry-spec>

Supported Configurations

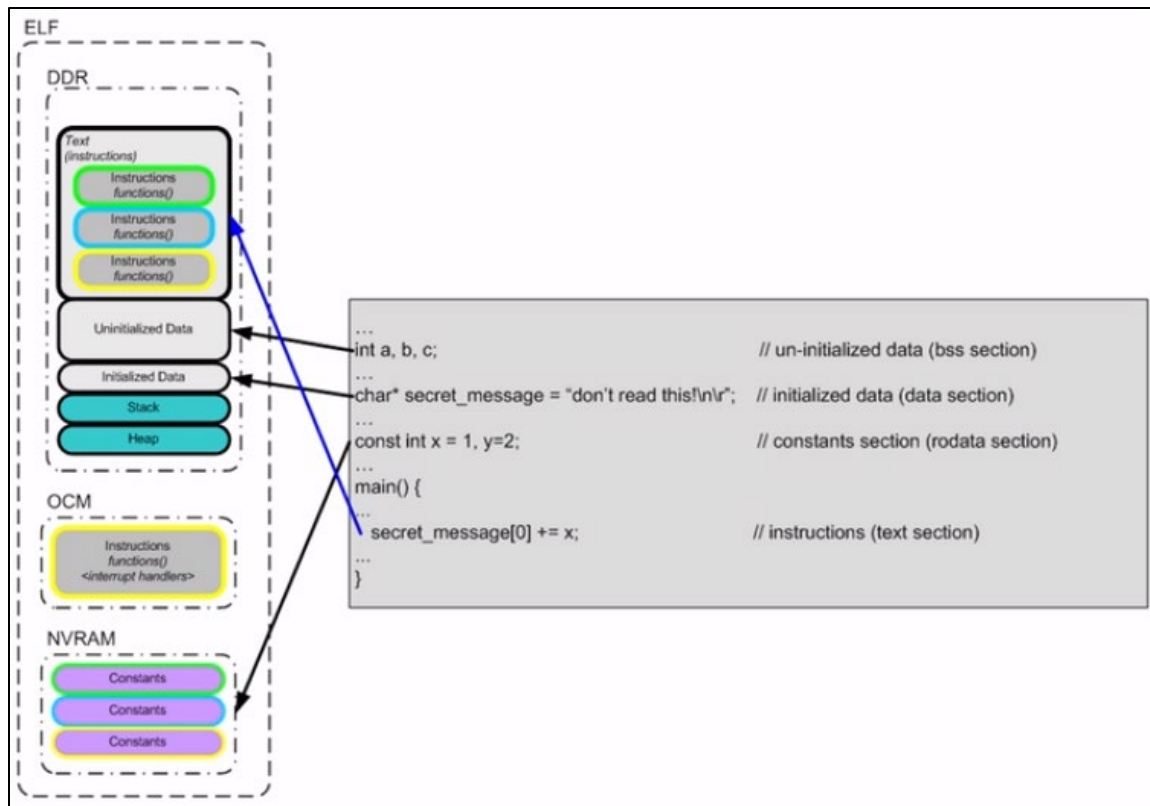
With the changes proposed here, we aim to support ECs with the following configurations:

- Internal mapped storage
 - External mapped storage (shared or dedicated)
 - External unmapped storage (shared or dedicated)
 - Code executed directly from mapped storage
 - Code copied to SRAM before use
-
- One RO image
 - One contiguous region of storage belonging to the EC, containing the RO image, that can be write protected
 - Up to one RW image
 - Up to one contiguous region of storage belonging to the EC, containing the RW image, that will not be write protected
 - Support for any number of chip-specific non-image data pieces (*NIDs*) as part of the storage regions (loaders, headers, etc - any piece of data on EC storage that isn't part of the RO or RW image)
 - Write protected and writable EC storage regions need not be mutually contiguous

Source: <https://www.chromium.org/chromium-os/ec-development/ec-image-geometry-spec>



Source: <https://www.xilinx.com/training/customer-training/using-linker-scripts.html> (2:20)



Source: <https://www.xilinx.com/training/customer-training/using-linker-scripts.html> (9:52)

Linking

Our linker scripts were originally written to support internal memory-mapped storage with only RO and RW images (no NIDs). The scripts will be extended to support the new configs defined above. For example, the linker will produce a unified image with the RW image at the appropriate offset, as defined by CONFIG_RW_STORAGE_OFF / CONFIG_RW_STORAGE_SIZE.

Source: <https://www.chromium.org/chromium-os/ec-development/ec-image-geometry-spec>

The microcontroller boot process starts by simply applying power to the system. Once the voltage rails stabilize, the microcontroller looks to the reset vector for the location in flash where the start-up instruction can be found. The reset vector is a special location within the flash memory

Source: <https://www.beningo.com/understanding-the-microcontroller-boot-process/>

The address that is stored at the reset vector is loaded by the microcontroller and the instructions that are contained there are then loaded and executed by the CPU. Now these first instructions aren't the start of main that the developer created. Instead, these are instructions on how to start-up the microcontroller.

The first thing that usually occurs is that the vector tables that are stored in flash are copied to RAM. They are copied from and to the location that is specified in the linker file at the time the executable program is created. One reason for copying the vector tables to RAM is that it is faster to execute from RAM than flash. This helps to decrease the latency of any interrupt calls within the system. Depending on the particular architecture of the microcontroller there may then be an instruction to update a vector table register so that the microcontroller knows where the start of the RAM table is.

Next the initialized data sections are copied into RAM. This is usually variables that are stored in the .data section of the linker. Examples of initialized data would be static, global and static local variables that have been provided with an initialization value during compile time. These are explicit definitions such as `int Var = 0x32;`.

Following the copy of the data section, the .bss section is also copied. The .bss section contains variables that are not initialized explicitly or that have been initialized to a value of zero. A simple example is that the variable `static int Var;` would be contained within this section.

Finally, the microcontroller will copy any RAM functions from flash to RAM. Once again it is sometimes worthwhile to execute certain functions out of RAM rather than flash due to the execution speed being slightly faster. These are functions usually decided upon by the developer and purposely placed there in the linker file prior to compiling the program.

Source: <https://www.beningo.com/understanding-the-microcontroller-boot-process/>

46. During initialization and booting of the EC (“micro-controller”), the executable version of the RO firmware (“boot code”) is executed first. The RO firmware then runs the executable version of the RW firmware (“first system code”). Accordingly, the boot code and system code are orderly executed to control booting of the embedded system.

EC Software Sync

It is important that the AP firmware (BIOS) and the EC firmware remain compatible through upgrades. At every* cold boot/reset of the EC

1. The EC boots its RO firmware, and powers on the AP.
2. The AP boots its RO firmware.
3. The AP verifies its RW firmware and jumps to it.
4. The EC computes a hash of its RW firmware.
5. The AP RW firmware contains a copy of the EC's RW firmware. The AP compares its hash with the EC's hash.
6. If they differ, the AP gives the EC the correct RW firmware, which the EC writes to its flash.
7. The EC jumps to its RW firmware.

There also are a few other tricks to ensure the EC isn't lying about its hash

*Normal mode, anyway. In recovery mode both AP and EC stay in their RO firmware

Source: https://www.coreboot.org/images/5/50/An_Open_Source_EC.pdf (Page 22).

Firmware Image

The Chrome OS firmware image has two main sections: Read-Only (RO) and Read-Write (RW). The RO firmware is set at the factory and cannot be updated after manufacturing. The RW firmware can be updated during Chrome OS auto-update (AU).

If a problem is found in RO firmware, Google creates an update and places it in the RW firmware. During the boot process, the RO firmware checks whether there is an update in the RW section and, if so, jumps to the RW update to execute the new boot code.

The RO firmware contains the following code:

- U-Boot, including the device tree for this system
- On x86 systems: coreboot
- Google Binary Block (GBB), which contains the following:
 - Recovery screen images
 - Public keys needed to verify the RW firmware
- Firmware ID (a string with the version number and device type)

The RW firmware contains two sections: A and B. Each section contains the following:

- U-Boot, including the device tree for this system (identical to the U-Boot images in RO firmware)
- VBlock, which contains the signatures used to verify the kernel before loading and running it
- Firmware ID
- Embedded Controller image
- Fmap, a data structure that describes the layout and contents of the SPI Flash. This structure is required by the Flashrom tool.

Source: <https://www.chromium.org/chromium-os/firmware-porting-guide/2-concepts?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1>

BASE	SIZE	SECTION	DESCRIPTION
0x000000	0x200000	SI_ALL	Descriptor + ME
0x200000	0x0f0000	RW_SECTION_A	Read-Write Firmware A
0x2f0000	0x0f0000	RW_SECTION_B	Read-Write Firmware B
0x3e0000	0x010000	RW_MRC_CACHE	Memory Training Cache
0x3f0000	0x004000	RW_ELOG	Event Log
0x3f4000	0x004000	RW_SHARED	Shared Data
0x3f8000	0x002000	RW_VPD	Read-Write VPD
0x400000	0x200000	RW_LEGACY	Legacy Firmware
0x600000	0x004000	RO_VPD	Read-Only VPD
0x610000	0x000800	FMAP	Flash Map
0x610800	0x000040	RO_FRID	RO Firmware ID
0x611000	0x0ef000	GBB	Google Binary Block
0x700000	0x100000	BOOT_STUB	Read-Only Firmware

Source: https://docs.google.com/presentation/d/1h-nsDGIQmYI2ldr95nYgLmyCYDgBIpJWSt9b7AqTZaw/pub?start=false&loop=false&delayms=3000&slide=id.g2b77a1dcf_1128 (Slide 33).

47. The Dell Latitude 5300 2-in-1 Chromebook Enterprise includes a connecting interface that is coupled to the micro-controller and further coupled to an external data storage device through a data transmission media. The external data storage device stores a second system software.

48. For example, the EC (“micro-controller”), which controls peripheral connections of the device like USB, Wi-Fi, etc., receives an updated version of the RW firmware (“second system software”) from the device’s Application Processor (AP). Specifically, the updated RW firmware (“second system software”) is read from the network server when the device is connected to the internet through a Wi-Fi network interface. Accordingly, the EC (“micro-controller”) is configured to be coupled to the network server (“external data storage device”) through the Wi-Fi network interface (“connecting interface”). The network server stores the RW

firmware update (“second system software”), which can be read by the device via the internet (“data transmission media”).

EC Software Sync

It is important that the AP firmware (BIOS) and the EC firmware remain compatible through upgrades. At every* cold boot/reset of the EC

1. The EC boots its RO firmware, and powers on the AP.
2. The AP boots its RO firmware.
3. The AP verifies its RW firmware and jumps to it.
4. The EC computes a hash of its RW firmware.
5. The AP RW firmware contains a copy of the EC's RW firmware. The AP compares its hash with the EC's hash.
6. If they differ, the AP gives the EC the correct RW firmware, which the EC writes to its flash.
7. The EC jumps to its RW firmware.

There also are a few other tricks to ensure the EC isn't lying about its hash

*Normal mode, anyway. In recovery mode both AP and EC stay in their RO firmware

Source: https://www.coreboot.org/images/5/50/An_Open_Source_EC.pdf (Page 22).

Google

Power Sequencing

- Each AP family has its own
 - Power states
 - Voltage regulators
 - Control GPIOs (both input and output)
 - Transition rules
 - Timing requirements
 - Trigger events
- The EC must manage and respond to all those requirements as the AP boots, sleeps, idles, or transitions between various subtle states.
- It must also ensure that certain peripherals are brought up and down accordingly (USB, WiFi, etc.)

Source: https://docs.google.com/presentation/d/1Xa_Z5SjW-soPvkugAR8_TEJFrJpzoZUa9HNR14Sjs8/pub?start=false&loop=false&delayms=3000&slide=id.g2bbed09ac_142 (Slide 21).

3. Verified Boot

Chromebook's startup is very different from Windows or Mac machines. When Chrome OS boots, it compares every component of the operating system with the current version verified by Google through the Internet. If there is a discrepancy, it will replace with the up-to-date version. Every time the Chromebook starts up, it does the self-check called "Verified Boot."

The self-check ensures Chrome OS in the right shape; it plays a fundamental role in Chromebook security mechanism.

- Drive automatic update: download new updates of Chrome OS when Verified Boot;
- Repair corrupted OS: take Chrome OS back if malware manages to escape the Sandbox;

Source: <https://www.keepds.com/tool/list?os=c>

Lightweight designs and HD touchscreens are paired with built in security and automatic backup on Google Drive.

- Auto updates¹
- Reimagined hardware
- Sleek and lightweight devices

Discover your Chromebook

Find yours

Shop by brand:

acer ASUS DELL Google hp

¹Requires internet connection. ^{**}This Google One membership offer (Offer) provides you with subscription benefits at no charge for a period of twelve months from the day you redee

Source: https://www.walmart.ca/en/electronics/laptops-computers/laptops-notebooks/chromebooks/google/N-1990+1000268?mtr=mdv_00439&icid=electronics_wmg_display_walmart_14hb_wk16_google_chromebook_en

Google rolls out security updates as soon as they're ready and applies them when a Chromebook boots up. In other words, during the boot sequence, Chrome OS checks to see if a new update is available. If yes, installs it without interrupting the user. At this point, your Internet has to be available.

Source: <https://www.keepds.com/tool/list?os=c>

If peer-to-peer (P2P) networking is available, devices can automatically update Chrome from nearby devices of the same model. This option reduces external network traffic. If P2P automatic updating fails or isn't possible on your network, devices update as usual. They either download the update from Google's servers or an intermediate web-caching proxy server.

Source: <https://support.google.com/chrome/a/answer/3168106?hl=en>

Firmware Image

The Chrome OS firmware image has two main sections: Read-Only (RO) and Read-Write (RW). The RO firmware is set at the factory and cannot be updated after manufacturing. The RW firmware can be updated during Chrome OS auto-update (AU).

If a problem is found in RO firmware, Google creates an update and places it in the RW firmware. During the boot process, the RO firmware checks whether there is an update in the RW section and, if so, jumps to the RW update to execute the new boot code.

The RO firmware contains the following code:

- U-Boot, including the device tree for this system
- On x86 systems: coreboot
- Google Binary Block (GBB), which contains the following:
 - Recovery screen images
 - Public keys needed to verify the RW firmware
- Firmware ID (a string with the version number and device type)

The RW firmware contains two sections: A and B. Each section contains the following:

- U-Boot, including the device tree for this system (identical to the U-Boot images in RO firmware)
- VBlock, which contains the signatures used to verify the kernel before loading and running it
- Firmware ID
- Embedded Controller image
- Fmap, a data structure that describes the layout and contents of the SPI Flash. This structure is required by the Flashrom tool.

Source: <https://www.chromium.org/chromium-os/firmware-porting-guide/2-concepts?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1>

49. The Dell Latitude 5300 2-in-1 Chromebook Enterprise includes boot code, which includes update agent interface programming (UAIP). The micro-controller is capable of executing the update agent interface programming to read the second system software from the external data storage device through the connecting interface before executing the system code.

50. During the boot process of the system, the executable version of the RO firmware (“boot code”) is executed before the RW firmware. And before the RW firmware (“first system code”) is executed, the EC initiates a Software Sync process by powering on and booting the device’s AP. The system verifies and, if needed, updates the RW firmware by comparing the version of the RW firmware. If there is an available update to the RW firmware (“second system software”), the update is sent to flash memory. Accordingly, the RO firmware (“boot code”) includes code or programming (“update agent interface programming”) to initiate the software sync process that updates the RW firmware.

EC Software Sync

It is important that the AP firmware (BIOS) and the EC firmware remain compatible through upgrades. At every* cold boot/reset of the EC

1. The EC boots its RO firmware, and powers on the AP.
2. The AP boots its RO firmware.
3. The AP verifies its RW firmware and jumps to it.
4. The EC computes a hash of its RW firmware.
5. The AP RW firmware contains a copy of the EC’s RW firmware. The AP compares its hash with the EC’s hash.
6. If they differ, the AP gives the EC the correct RW firmware, which the EC writes to its flash.
7. The EC jumps to its RW firmware.

There also are a few other tricks to ensure the EC isn’t lying about its hash

*Normal mode, anyway. In recovery mode both AP and EC stay in their RO firmware

Source: https://www.coreboot.org/images/5/50/An_Open_Source_EC.pdf (Page 22).

U-Boot and Embedded Controller

U-Boot performs the device initialization for the system, as follows:

1. U-Boot calls the Vblnit() function to start verified boot.
2. In the simplest case, the boot consists of one step: running the code located in the Read-Only (RO) section of the firmware. During this process, U-Boot checks whether there are any updates in the Read/Write (RW) section of the firmware. If updates are present, U-Boot loads and runs RW firmware.
3. The main firmware performs a software sync that checks whether the EC code needs to be updated. If so, the update is sent over I2C, SPI, or LPC to the EC.
4. Once the EC code has been sync’ed, execution jumps to the EC code in the RW firmware.
5. The kernel is loaded and verified.
6. The correct device tree file is selected for the system (ARM only).
7. In addition, the kernel contains command line information that U-Boot picks up and passes to the kernel for use during boot. The command line tells the kernel which device to boot from (eMMC, SD, USB) and also contains the Verity parameters that are passed in to the kernel at boot time.
8. The system boots the kernel, which uses the root hash (contained in the Verity parameters) to open the root filesystem.
9. The system initializes user space and runs X and Chrome.

Source: <https://www.chromium.org/chromium-os/firmware-porting-guide/2-concepts?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1>

3. Verified Boot

Chromebook's startup is very different from Windows or Mac machines. When Chrome OS boots, it compares every component of the operating system with the current version verified by Google through the Internet. If there is a discrepancy, it will replace with the up-to-date version. Every time the Chromebook starts up, it does the self-check called "Verified Boot."

The self-check ensures Chrome OS in the right shape; it plays a fundamental role in Chromebook security mechanism.

- Drive automatic update: download new updates of Chrome OS when Verified Boot;
- Repair corrupted OS: take Chrome OS back if malware manages to escape the Sandbox;

Source: <https://www.keepds.com/tool/list?os=c>

Terminology

RO and RW

MCUs running the EC code have read-only (RO) and read-write (RW) firmware. Coming out of reset, the MCU boots into its RO firmware.

In the case of the EC, the RO firmware boots the host and asks it verify a hash of the RW firmware (software sync). If the RW firmware is invalid, it is updated from a copy in the host's RW firmware.

Source:

https://chromium.googlesource.com/chromiumos/platform/ec/+master/docs/write_protection.md

A feature called "Software Sync" keeps a copy of the read-write (RW) EC firmware in the RW part of the system firmware image. At boot, if the RW EC firmware doesn't match the copy in the system firmware, the EC's RW section is re-flashed.

Source: <https://chromium.googlesource.com/chromiumos/platform/ec/>

51. During the Verified Boot process, the system checks for updates to the RW firmware. The RO firmware includes programming that verifies and compares the version of the RW firmware on the device with the version of the RW firmware on the network server. The RO firmware further includes programming that updates the RW firmware based on that

comparison. The Software Sync process is a part of the Verified Boot process. It includes programming that verifies and compares version of the RW firmware.

Chrome EC

- Embedded Controllers are vital but closed
- Chrome EC is open source
 - chromiumos/platform/ec.git
- Chrome EC is designed for security
 - RO and RW regions
 - RW update is signed and handled by host firmware
 - EC Software Sync is part of Verified Boot
- Support for different ARM SOCs
 - Texas Instruments Stellaris Cortex-M4
 - ST Micro STM32 Cortex-M3
 - More in progress...

Source: [https://docs.google.com/presentation/d/1h-](https://docs.google.com/presentation/d/1h-nsDGIQmYI21dr95nYgLmyCYDgBIpJWSt9b7AqTZaw/pub?start=false&loop=false&delayms=3000&slide=id.g2b77a1dcf_298)

[nsDGIQmYI21dr95nYgLmyCYDgBIpJWSt9b7AqTZaw/pub?start=false&loop=false&delayms=3000&slide=id.g2b77a1dcf_298](https://docs.google.com/presentation/d/1h-nsDGIQmYI21dr95nYgLmyCYDgBIpJWSt9b7AqTZaw/pub?start=false&loop=false&delayms=3000&slide=id.g2b77a1dcf_298) (Slide 29).

```
VbError_t VbEcSoftwareSync(VbCommonParams *cparams)
{
    VbSharedDataHeader *shared =
        (VbSharedDataHeader *)cparams->shared_data_blob;
    int in_rw = 0;
    int rv;
    const uint8_t *ec_hash = NULL;
    int ec_hash_size;
    const uint8_t *rw_hash = NULL;
    int rw_hash_size;
    const uint8_t *expected = NULL;
    int expected_size;
    uint8_t expected_hash[SHA256_DIGEST_SIZE];
    int need_update = 0;
    int i;

    /* Determine whether the EC is in RO or RW */
    rv = VbExEcRunningRW(&in_rw);
```


Source: https://chromium.googlesource.com/chromiumos/platform/vboot_reference/+/factory-spring-4262.B/firmware/lib/vboot_api_kernel.c

```
/* Get hash of EC-RW */
rv = VbExEcHashRW(&ec_hash, &ec_hash_size);
if (rv) {
    VBDEBUG(("VbEcSoftwareSync() - "
            "VbExEcHashRW() returned %d\n", rv));
    VbSetRecoveryRequest(VBNV_RECOVERY_EC_HASH_FAILED);
    return VBERROR_EC_REBOOT_TO_RO_REQUIRED;
}
if (ec_hash_size != SHA256_DIGEST_SIZE) {
    VBDEBUG(("VbEcSoftwareSync() - "
            "VbExEcHashRW() says size %d, not %d\n",
            ec_hash_size, SHA256_DIGEST_SIZE));
    VbSetRecoveryRequest(VBNV_RECOVERY_EC_HASH_SIZE);
    return VBERROR_EC_REBOOT_TO_RO_REQUIRED;
}

VBDEBUG(("EC hash:"));
for (i = 0; i < SHA256_DIGEST_SIZE; i++)
    VBDEBUG(("%"02x", ec_hash[i]));
VBDEBUG(("\\n"));
```

Source: https://chromium.googlesource.com/chromiumos/platform/vboot_reference/+/factory-spring-4262.B/firmware/lib/vboot_api_kernel.c

```

/*
 * Get expected EC-RW image if we're sure we need to update (because the
 * expected hash didn't match the EC) or we still don't know (because
 * there was no expected hash and we need the image to compute one
 * ourselves).
 */
if (need_update || !rw_hash) {
    /* Get expected EC-RW image */
    rv = VbExEcGetExpectedRW(shared->firmware_index ?
                            VB_SELECT_FIRMWARE_B :
                            VB_SELECT_FIRMWARE_A,
                            &expected, &expected_size);

    if (rv) {
        VBDEBUG(("VbEcSoftwareSync() - "
                "VbExEcGetExpectedRW() returned %d\n", rv));
        VbSetRecoveryRequest(VBNV_RECOVERY_EC_EXPECTED_IMAGE);
        return VBERROR_EC_REBOOT_TO_RO_REQUIRED;
    }
    VBDEBUG(("VbEcSoftwareSync() - expected len = %d\n",
            expected_size));

    /* Hash expected image */
    internal_SHA256(expected, expected_size, expected_hash);
    VBDEBUG(("Computed hash of expected image:"));
    for (i = 0; i < SHA256_DIGEST_SIZE; i++)
        VBDEBUG(("%02x", expected_hash[i]));
    VBDEBUG((" \n"));
}

```

Source: https://chromium.googlesource.com/chromiumos/platform/vboot_reference/+/factory-spring-4262.B/firmware/lib/vboot_api_kernel.c

```

/*
 * We need to update, but the expected EC image doesn't match
 * the expected EC hash we were given.
 */
VBDEBUG(("VbEcSoftwareSync() - "
        "VbExEcGetExpectedRW() returned %d\n", rv));
VbSetRecoveryRequest(VBNV_RECOVERY_EC_HASH_MISMATCH);
return VBERROR_EC_REBOOT_TO_RO_REQUIRED;

```

Source: https://chromium.googlesource.com/chromiumos/platform/vboot_reference/+/factory-spring-4262.B/firmware/lib/vboot_api_kernel.c

```

/* Update EC if necessary */
if (need_update) {
    VBDEBUG(("VbEcSoftwareSync() updating EC-RW...\n"));

    if (shared->flags & VBSD_EC_SLOW_UPDATE) {
        VBDEBUG(("VbEcSoftwareSync() - "
            "EC is slow. Show WAIT screen.\n"));

        /*
         * FIXME(crosbug.com/p/12257): Ensure the VGA Option
         * ROM is loaded!
         */
        VbDisplayScreen(cparams, VB_SCREEN_WAIT, 0, &vnc);
    }

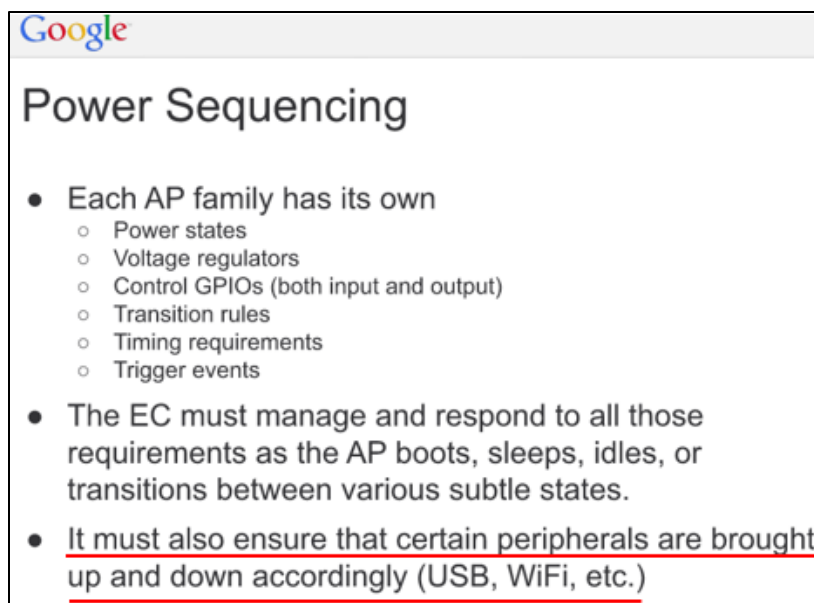
    rv = VbExecUpdateRW(expected, expected_size);
    if (rv == VBERROR_EC_REBOOT_TO_RO_REQUIRED) {
        /*
         * Reboot required. May need to unprotect RW before
         * updating, or may need to reboot after RW updated.
         * Either way, it's not an error requiring recovery
         * mode.
         */
        VBDEBUG(("VbEcSoftwareSync() - "
            "VbExecUpdateRW() needs reboot\n"));
    }
    return rv;
}

```

Source: https://chromium.googlesource.com/chromiumos/platform/vboot_reference/+/factory-spring-4262.B/firmware/lib/vboot_api_kernel.c

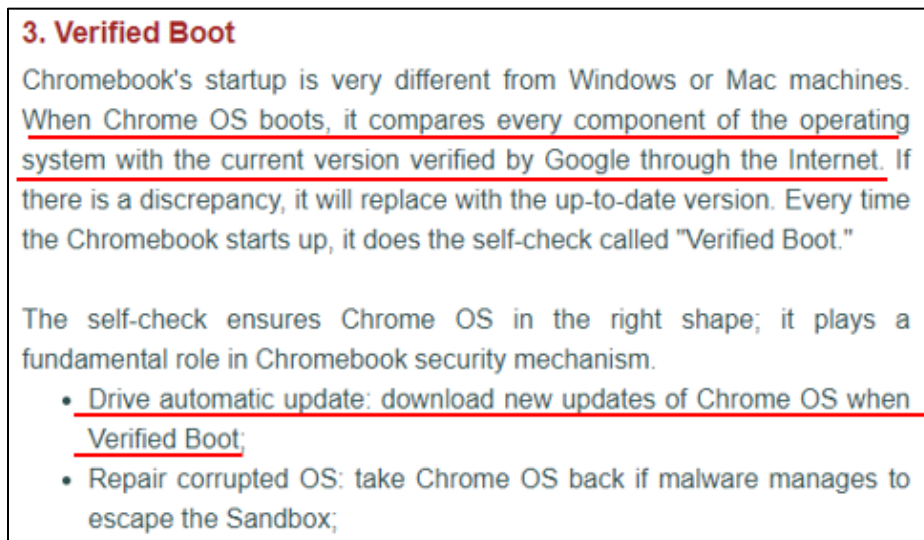
52. The EC controls peripheral connections like USB, Wi-Fi, etc. During the Verified Boot process, the device receives updates of the RW firmware from the network server (“external data storage device”) when connected to the internet through Wi-Fi, for example. The RW firmware is updated during the Verified Boot process.

53. The EC initiates the process of updating the RW firmware by activating the Wi-Fi network interface, connecting to the internet, and starting the Software Sync process. Accordingly, the system includes code or programming (“update agent interface programming”) to read the RW firmware update (“second system software”) from the network server (“external data storage device”) before executing the current version of the RW firmware (“first system code”).



The image is a screenshot of a Google presentation slide. At the top left, the Google logo is visible. The slide title is "Power Sequencing". Below the title, there are three main bullet points. The first bullet point is "Each AP family has its own", followed by a sub-list of six items: "Power states", "Voltage regulators", "Control GPIOs (both input and output)", "Transition rules", "Timing requirements", and "Trigger events". The second bullet point is "The EC must manage and respond to all those requirements as the AP boots, sleeps, idles, or transitions between various subtle states." The third bullet point is "It must also ensure that certain peripherals are brought up and down accordingly (USB, WiFi, etc.)", which is underlined in red.

Source: https://docs.google.com/presentation/d/1Xa_Z5SjW-soPvkugAR8_TEJFrJpzoZUa9HNR14Sjs8/pub?start=false&loop=false&delayms=3000&slide=id.g2bbed09ac_142 (Slide 21).




The image is a screenshot of a presentation slide titled "3. Verified Boot". The text describes Chromebook's startup process, stating that it compares every component of the operating system with the current version verified by Google through the Internet. It mentions that if there is a discrepancy, it will replace it with the up-to-date version. The slide also notes that every time the Chromebook starts up, it performs a self-check called "Verified Boot." Below this, it states that the self-check ensures Chrome OS is in the right shape and plays a fundamental role in Chromebook security. At the bottom, there are two bullet points: "Drive automatic update: download new updates of Chrome OS when Verified Boot;" and "Repair corrupted OS: take Chrome OS back if malware manages to escape the Sandbox;".

Source: <https://www.keepds.com/tool/list?os=c>

Lightweight designs and HD touchscreens are paired with built in security and automatic backup on Google Drive.

- Auto updates¹
- Reimagined hardware
- Sleek and lightweight devices



Discover your Chromebook

Find yours

Shop by brand:

acer ASUS DELL Google hp

¹Requires internet connection. ²This Google One membership offer (Offer) provides you with subscription benefits at no charge for a period of twelve months from the day you redeem.

Source: https://www.walmart.ca/en/electronics/laptops-computers/laptops-notebooks/chromebooks/google/N-1990+1000268?mtr=mdv_00439&icid=electronics_wmg_display_walmart_l4hb_wk16_google_chromebook_en

Google rolls out security updates as soon as they're ready and applies them when a Chromebook boots up. In other words, during the boot sequence, Chrome OS checks to see if a new update is available. If yes, installs it without interrupting the user. At this point, your Internet has to be available.

Source: <https://www.keepds.com/tool/list?os=c>

If peer-to-peer (P2P) networking is available, devices can automatically update Chrome from nearby devices of the same model. This option reduces external network traffic. If P2P automatic updating fails or isn't possible on your network, devices update as usual. They either download the update from Google's servers or an intermediate web-caching proxy server.

Source: <https://support.google.com/chrome/a/answer/3168106?hl=en>

Firmware Image

The Chrome OS firmware image has two main sections: Read-Only (RO) and Read-Write (RW). The RO firmware is set at the factory and cannot be updated after manufacturing. The RW firmware can be updated during Chrome OS auto-update (AU).

If a problem is found in RO firmware, Google creates an update and places it in the RW firmware. During the boot process, the RO firmware checks whether there is an update in the RW section and, if so, jumps to the RW update to execute the new boot code.

The RO firmware contains the following code:

- U-Boot, including the device tree for this system
- On x86 systems: coreboot
- Google Binary Block (GBB), which contains the following:
 - Recovery screen images
 - Public keys needed to verify the RW firmware
- Firmware ID (a string with the version number and device type)

The RW firmware contains two sections: A and B. Each section contains the following:

- U-Boot, including the device tree for this system (identical to the U-Boot images in RO firmware)
- VBlock, which contains the signatures used to verify the kernel before loading and running it
- Firmware ID
- Embedded Controller image
- Fmap, a data structure that describes the layout and contents of the SPI Flash. This structure is required by the Flashrom tool.

Source: <https://www.chromium.org/chromium-os/firmware-porting-guide/2-concepts?tmpl=%2Fsystem%2Fapp%2Ftemplates%2Fprint%2F&showPrintDialog=1>

54. Dell has had knowledge of the '612 Patent at least as of the date when it was notified of the filing of this action.

55. Liberty Patents has been damaged as a result of the infringing conduct by Dell alleged above. Thus, Dell is liable to Liberty Patents in an amount that adequately compensates it for such infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

56. Liberty Patents and/or its predecessors-in-interest have satisfied all statutory obligations required to collect pre-filing damages for the full period allowed by law for infringement of the '612 Patent.

ADDITIONAL ALLEGATIONS REGARDING INFRINGEMENT

57. Dell has also indirectly infringed the '573 Patent and the '612 Patent by inducing others to directly infringe the '573 Patent and the '612 Patent. Dell has induced the end-users, Dell's customers, to directly infringe (literally and/or under the doctrine of equivalents) the '573 Patent and the '612 Patent by using the accused products.

58. Dell took active steps, directly and/or through contractual relationships with others, with the specific intent to cause them to use the accused products in a manner that infringes one or more claims of the patents-in-suit, including, for example, claim 13 of the '573 Patent and claim 1 of the '612 Patent.

59. Such steps by Dell included, among other things, advising or directing customers and end-users to use the accused products in an infringing manner; advertising and promoting the use of the accused products in an infringing manner; and/or distributing instructions that guide users to use the accused products in an infringing manner.

60. Dell performed these steps, which constitute induced infringement, with the knowledge of the '573 Patent and the '612 Patent and with the knowledge that the induced acts constitute infringement.

61. Dell was and is aware that the normal and customary use of the accused products by Dell's customers would infringe the '573 Patent and the '612 Patent. Dell's inducement is ongoing.

62. Dell has also induced its affiliates, or third-party manufacturers, shippers, distributors, retailers, or other persons acting on its or their affiliates' behalf, to directly infringe (literally and/or under the doctrine of equivalents) the '573 Patent and '612 Patent by importing, selling or offering to sell the accused products, including, for example, Amazon, Best Buy, Office Depot, Sam's Club, Staples, Walmart, and others.

63. Dell has a significant role in placing the accused products in the stream of commerce with the expectation and knowledge that they will be purchased by consumers in Texas and elsewhere in the United States.

64. Dell purposefully directs or controls the making of accused products and their shipment to the United States, using established distribution channels, for sale in Texas and elsewhere within the United States.

65. Dell purposefully directs or controls the sale of the accused products into established United States distribution channels, including sales to nationwide retailers. Dell directs or controls the sale of the accused products online and in nationwide retailers, including for sale in Texas and elsewhere in the United States, and expects and intends that the accused products will be so sold.

66. Dell purposefully places the accused products—whether by itself or through subsidiaries, affiliates, or third parties—into an international supply chain, knowing that the accused products will be sold in the United States, including Texas. Therefore, Dell also facilitates the sale of the accused products in Texas.

67. Dell took active steps, directly and/or through contractual relationships with others, with the specific intent to cause such persons to import, sell, or offer to sell the accused products in a manner that infringes one or more claims of the patents-in-suit, including, for example, claim 13 of the '573 Patent and claim 1 of the '612 Patent.

68. Such steps by Dell included, among other things, making or selling the accused products outside of the United States for importation into or sale in the United States, or knowing that such importation or sale would occur; and directing, facilitating, or influencing its affiliates, or third-party manufacturers, shippers, distributors, retailers, or other persons acting on its or their behalf, to import, sell, or offer to sell the accused products in an infringing manner.

69. Dell performed these steps, which constitute induced infringement, with the knowledge of the '573 Patent and the '612 Patent and with the knowledge that the induced acts would constitute infringement.

70. Dell performed such steps in order to profit from the eventual sale of the accused products in the United States.

71. Dell's inducement is ongoing.

72. Dell has also indirectly infringed by contributing to the infringement of the '573 Patent and the '612 Patent. Dell has contributed to the direct infringement of the '573 Patent and the '612 Patent by the end-user of the accused products.

73. The accused products have special features that are specially designed to be used in an infringing way and that have no substantial uses other than ones that infringe the '573 Patent and the '612 Patent, for example, claim 13 of the '573 Patent and claim 1 of the '612 Patent.

74. The special features include, for example, power distribution and power management techniques used in a manner that infringes the '573 Patent and retrieving automatic software updates in an embedded system used in a manner that infringes the '612 Patent.

75. These special features constitute a material part of the invention of one or more of the claims of the '573 Patent and the '612 Patent and are not staple articles of commerce suitable for substantial non-infringing use.

76. Dell's contributory infringement is ongoing.

77. Dell has had actual knowledge of the '573 Patent and the '612 Patent at least as of the date when it was notified of the filing of this action. Since at least that time, Dell has known the scope of the claims of the '573 Patent and the '612 Patent; the products that practice the '573

Patent and the '612 Patent; and that Liberty Patents is the owner of the '573 Patent and the '612 Patent.

78. By the time of trial, Dell will have known and intended (since receiving such notice) that its continued actions would infringe and actively induce and contribute to the infringement of one or more claims of the '573 Patent and the '612 Patent.

79. Furthermore, Dell has a policy or practice of not reviewing the patents of others (including instructing its employees to not review the patents of others), and thus has been willfully blind of Liberty Patents' patent rights. *See, e.g.*, M. Lemley, "Ignoring Patents," 2008 Mich. St. L. Rev. 19 (2008).

80. Dell's actions are at least objectively reckless as to the risk of infringing valid patents, and this objective risk was either known or should have been known by Dell. Dell has knowledge of the '573 Patent and the '612 Patent.

81. Dell's customers have infringed the '573 Patent and the '612 Patent, and Dell has encouraged its customers' infringement.

82. Dell's direct and indirect infringement of the '573 Patent and the '612 Patent has been, and/or continues to be willful, intentional, deliberate, and/or in conscious disregard of Liberty Patents' rights under the patents-in-suit.

83. Liberty Patents has been damaged as a result of Dell's infringing conduct alleged above. Thus, Dell is liable to Liberty Patents in an amount that adequately compensates it for such infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

JURY DEMAND

Liberty Patents hereby requests a trial by jury on all issues so triable by right.

PRAYER FOR RELIEF

Liberty Patents requests that the Court find in its favor and against Dell, and that the Court grant Liberty Patents the following relief:

a. Judgment that one or more claims of the '573 Patent and the '612 Patent have been infringed, either literally and/or under the doctrine of equivalents, by Dell and/or all others acting in concert therewith;

b. A permanent injunction enjoining Dell and its officers, directors, agents, servants, affiliates, employees, divisions, branches, subsidiaries, parents, and all others acting in concert therewith from infringement of the '573 Patent and the '612 Patent; or, in the alternative, an award of a reasonable ongoing royalty for future infringement of the '573 Patent and the '612 Patent by such entities;

c. Judgment that Dell account for and pay to Liberty Patents all damages to and costs incurred by Liberty Patents because of Dell's infringing activities and other conduct complained of herein, including an award of all increased damages to which Liberty Patents is entitled under 35 U.S.C. § 284;

d. That Liberty Patents be granted pre-judgment and post-judgment interest on the damages caused by Dell's infringing activities and other conduct complained of herein;

e. That this Court declare this an exceptional case and award Liberty Patents its reasonable attorney's fees and costs in accordance with 35 U.S.C. § 285; and

f. That Liberty Patents be granted such other and further relief as the Court may deem just and proper under the circumstances.

Dated: October 12, 2020

Respectfully submitted,

/s/ Zachariah S. Harrington

Matthew J. Antonelli

Texas Bar No. 24068432

matt@ahtlawfirm.com

Zachariah S. Harrington

Texas Bar No. 24057886

zac@ahtlawfirm.com

Larry D. Thompson, Jr.

Texas Bar No. 24051428

larry@ahtlawfirm.com

Christopher Ryan Pinckney

Texas Bar No. 24067819

ryan@ahtlawfirm.com

Rehan M. Safiullah

Texas Bar No. 24066017

rehan@ahtlawfirm.com

ANTONELLI, HARRINGTON

& THOMPSON LLP

4306 Yoakum Blvd., Ste. 450

Houston, TX 77006

(713) 581-3000

Stafford Davis

State Bar No. 24054605

sdavis@stafforddavisfirm.com

Catherine Bartles

Texas Bar No. 24104849

cbartles@stafforddavisfirm.com

THE STAFFORD DAVIS FIRM

815 South Broadway Avenue

Tyler, Texas 75701

(903) 593-7000

(903) 705-7369 fax

Attorneys for Liberty Patents, LLC