IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

LIBERTY PATENTS, LLC, Plaintiff, v. PANASONIC CORPORATION OF NORTH AMERICA, Defendant.	CIVIL ACTION NO. 2:20-cv-291 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 Defendant Panasonic Corporation of North America ("Panasonic" or "Defendant"), 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.

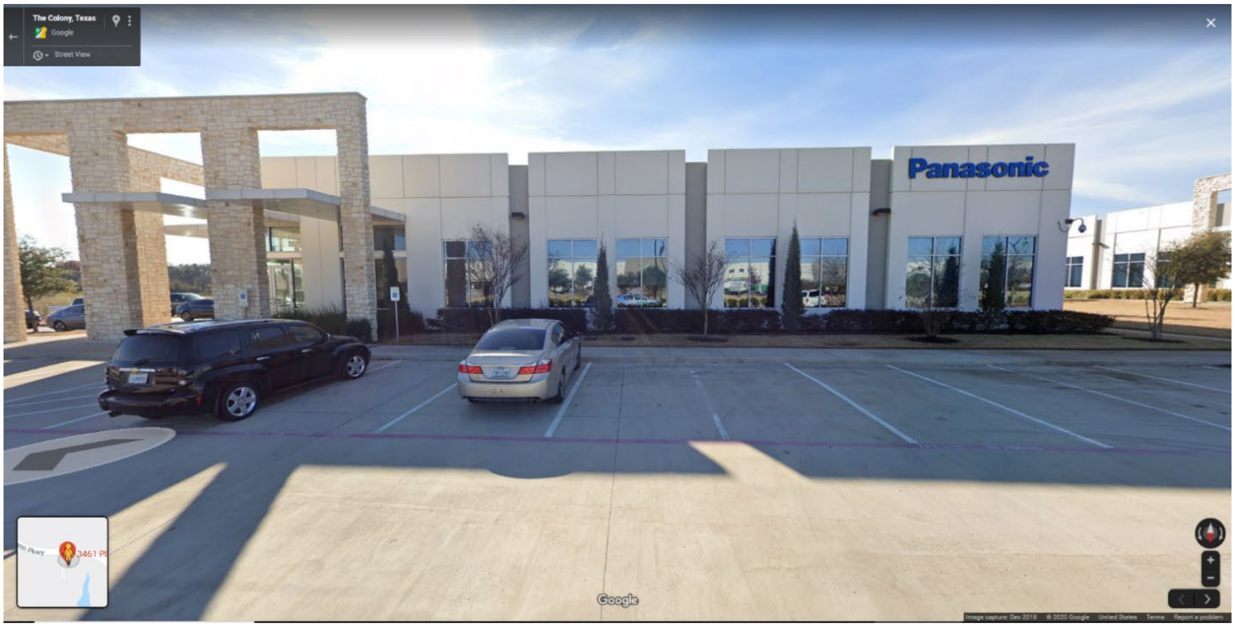
2. Defendant Panasonic Corporation of North America is a corporation duly organized and existing under the laws of Delaware. Panasonic Corporation of North America may be served through its registered agent, CT Corporation System, at 1999 Bryan St., Suite 900, Dallas, TX 75201.

JURISDICTION AND VENUE

3. 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).

4. This Court has personal jurisdiction over Panasonic pursuant to due process and/or the Texas Long Arm Statute because, *inter alia*, (i) Panasonic has done and continues to do business in Texas; (ii) Panasonic 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 in Texas; and (iii) Panasonic is registered to do business in Texas.

5. Venue is proper in this district pursuant to 28 U.S.C. § 1400(b). Venue is further proper because Panasonic has committed and continues 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. Panasonic also has regular and established places of business in this district, including at least at 3461 Plano Pkwy, The Colony, Texas 75056 in Denton County (depicted below).


The Colony, Texas
Google
Street View
Panasonic
3461 Pl
Google
Image capture: Dec 2018 © 2020 Google United States Terms Report a problem

Source: https://goo.gl/maps/yYJ4kJ2H2yEE76wZ6.

BACKGROUND

6. The three 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 more efficient handling of computer instructions for faster processing, better power distribution and power management, and a better process for retrieving automatic software updates.

7. U.S. Patent No. 6,535,959 ("the '959 Patent") discloses a processor that includes an instruction cache. The instruction cache is a set-associative cache that comprises multiple blocks. Claim 1 of the '959 patent is directed to a processor that generates a power reduction signal, which indicates whether the subsequent instruction to be executed resides in the same block of the instruction cache as the current instruction that is being executed. This advantageously allows, for example, the processor to read consecutive instructions (or

instructions that are in the same block) quickly, without multiple additional steps. The novel system results in a processor with increased operating speed and decreased power consumption.

8. The invention described in the ’959 Patent was the result of research conducted by two inventors at Conexant Systems, Inc., which was—at the time—the world’s largest, standalone communications-IC company. Conexant, itself, was a spin-off from the semiconductor division of the well-known and well-regarded Rockwell International Corp. Conexant was known as a leading supplier of innovative semiconductor solutions for imaging, audio, embedded modem, and video surveillance applications.[1] Recently, Conexant was acquired by Synaptics, the leading developer of human interface solutions for over $300 million. Since its formation, Conexant has been an innovator in the semiconductor field (and others) with more than a thousand patents assigned to it.

9. The ’959 Patent has been cited by multiple technology companies—as recently as 2017—including, Apple, Fujitsu, IBM, Honeywell, Intel, Matsushita, Oracle, and Samsung.

10. U.S. Patent No. 6,920,573 (“the ’573 Patent”) generally relates to a system for conserving energy in electronic systems. Specifically, the system 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-

[1] *See* Conexant’s Audio Solution Named CES Innovations 2011 Awards Honoree, BUSINESS WIRE (Nov. 9, 2010), https://www.businesswire.com/news/home/20101109005618/en/Conexant%E2%80%99s-Audio-Solution-Named-CES-Innovations-2011.

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.

11. 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 Google, Hewlett-Packard, Intel, Matsushita, Microsoft, NVIDIA, Sony, and Transmeta.

12. 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.

13. 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.

14. 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 Google, IBM, and Texas Instruments.

COUNT I

DIRECT INFRINGEMENT OF U.S. PATENT NO. 6,535,959

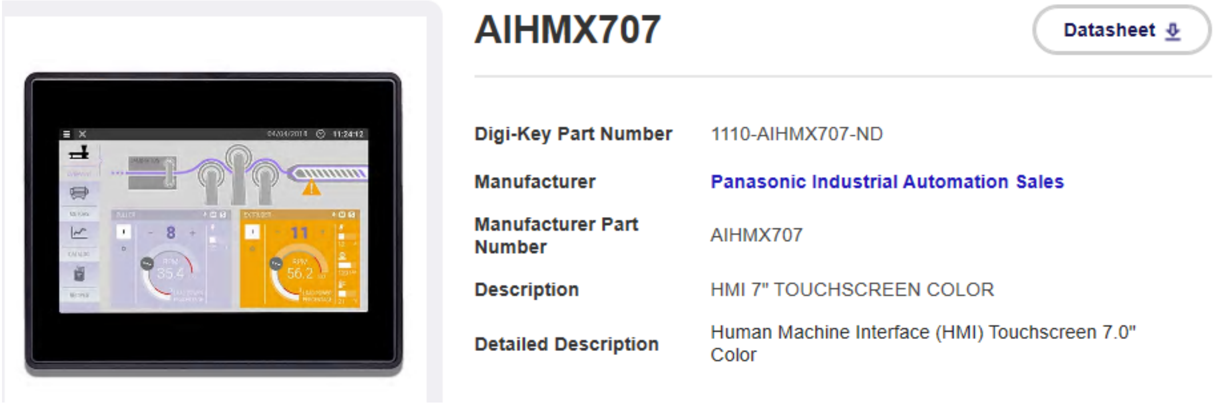
15. On March 18, 2003, the ’959 Patent was duly and legally issued by the United States Patent and Trademark Office for an invention entitled “Circuit and Method for Reducing Power Consumption in an Instruction Cache.”

16. Liberty Patents is the owner of the ’959 Patent, with all substantive rights in and to that patent, including the sole and exclusive right to prosecute this action and enforce the ’959 Patent against infringers, and to collect damages for all relevant times.

17. Panasonic made, had made, used, imported, provided, supplied, distributed, sold, and/or offered for sale products and/or systems including, for example, its Panasonic HMx707 HMI Terminal (AIHMX707), and other products that include processors with the capability to ignore reading the tag field when a sequential instruction is to be loaded[2] (processors such as the

[2] *See, e.g.*, Panasonic TC-43DS630, TC-49DS630, TX-40DS630, TX-50DS630, TX-32DS600, TC-40DS600 (31.5"), TC-40DS600, TC-49DS600, TX-24DS503, TX-32DS503, TX-40DS503, TX-49DS503, TX-55DS503, TX-24DS500, TX-32DS500, TX-40DS500, TX-49DS500, TX-55DS500, TX-40DSU501, TX-49DSU501, TX-55DSU501, TX-40DS400, TX-40DSU401;

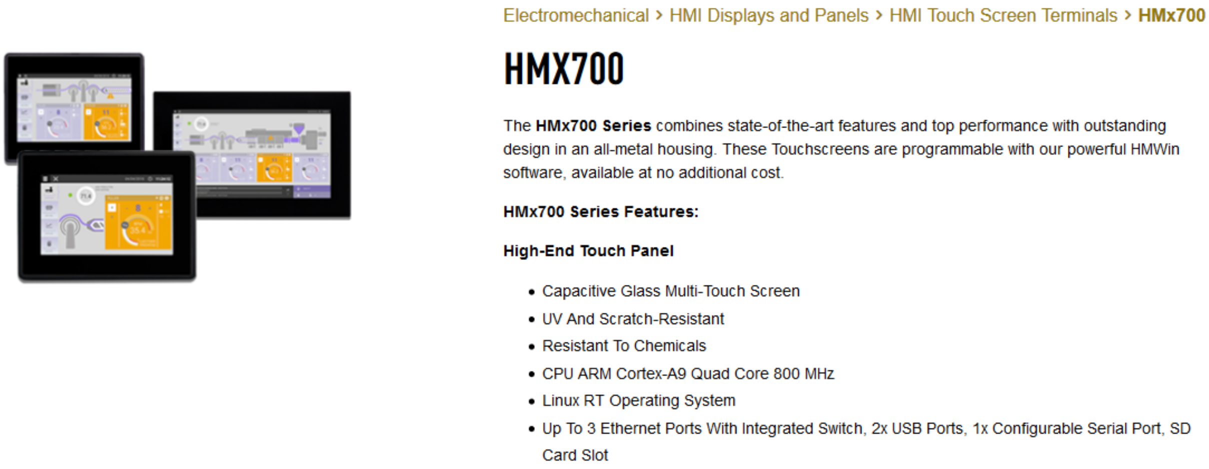
ARM Cortex-A72, Cortex-A57, Cortex-A15, Cortex-A9, Cortex-R5, Cortex-R4, ARM11, etc.) ("accused products"):



AIHMX707

Digi-Key Part Number	1110-AIHMX707-ND
Manufacturer	Panasonic Industrial Automation Sales
Manufacturer Part Number	AIHMX707
Description	HMI 7" TOUCHSCREEN COLOR
Detailed Description	Human Machine Interface (HMI) Touchscreen 7.0" Color

Source: www.digikey.com/product-detail/en/panasonic-industrial-automation-sales/AIHMX707/1110-AIHMX707-ND/11656288



Electromechanical > HMI Displays and Panels > HMI Touch Screen Terminals > HMx700

HMX700

The **HMx700 Series** combines state-of-the-art features and top performance with outstanding design in an all-metal housing. These Touchscreens are programmable with our powerful HMWin software, available at no additional cost.

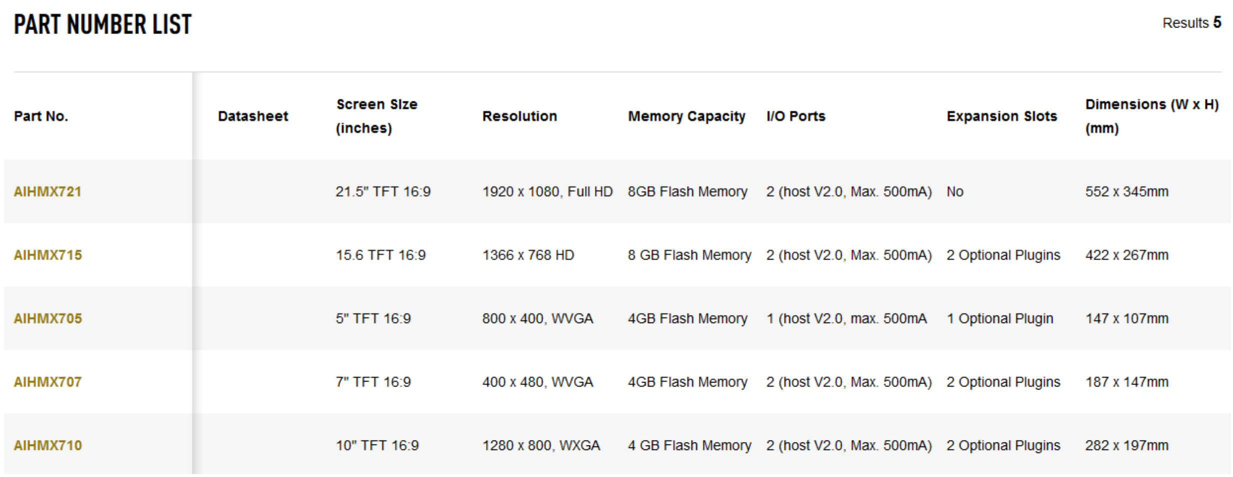
HMx700 Series Features:

High-End Touch Panel

- Capacitive Glass Multi-Touch Screen
- UV And Scratch-Resistant
- Resistant To Chemicals
- CPU ARM Cortex-A9 Quad Core 800 MHz
- Linux RT Operating System
- Up To 3 Ethernet Ports With Integrated Switch, 2x USB Ports, 1x Configurable Serial Port, SD Card Slot

Source: na.industrial.panasonic.com/products/electromechanical/hmi-displays-and-panels/lineup/hmi-touch-screen-terminals/series/133483

Panasonic JT-B1; HMX700 Series (HMx707, HMx710, HMx715, HMx720); Panasonic UniPhier MN2WS0220.



PART NUMBER LIST

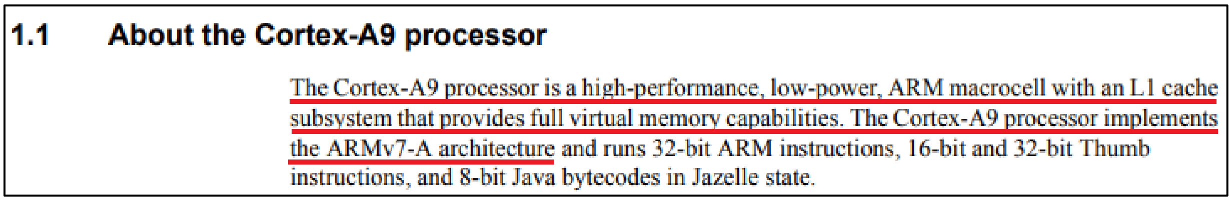
Results **5**

Part No.	Datasheet	Screen Size (inches)	Resolution	Memory Capacity	I/O Ports	Expansion Slots	Dimensions (W x H) (mm)
AIHMX721		21.5" TFT 16:9	1920 x 1080, Full HD	8GB Flash Memory	2 (host V2.0, Max. 500mA)	No	552 x 345mm
AIHMX715		15.6 TFT 16:9	1366 x 768 HD	8 GB Flash Memory	2 (host V2.0, Max. 500mA)	2 Optional Plugins	422 x 267mm
AIHMX705		5" TFT 16:9	800 x 400, WVGA	4GB Flash Memory	1 (host V2.0, max. 500mA	1 Optional Plugin	147 x 107mm
AIHMX707		7" TFT 16:9	400 x 480, WVGA	4GB Flash Memory	2 (host V2.0, Max. 500mA)	2 Optional Plugins	187 x 147mm
AIHMX710		10" TFT 16:9	1280 x 800, WXGA	4 GB Flash Memory	2 (host V2.0, Max. 500mA)	2 Optional Plugins	282 x 197mm

Source: na.industrial.panasonic.com/products/electromechanical/hmi-displays-and-panels/lineup/hmi-touch-screen-terminals/series/133483

18. By doing so, Panasonic has directly infringed (literally and/or under the doctrine of equivalents) at least Claim 1 of the '959 Patent. Panasonic's infringement in this regard is ongoing.

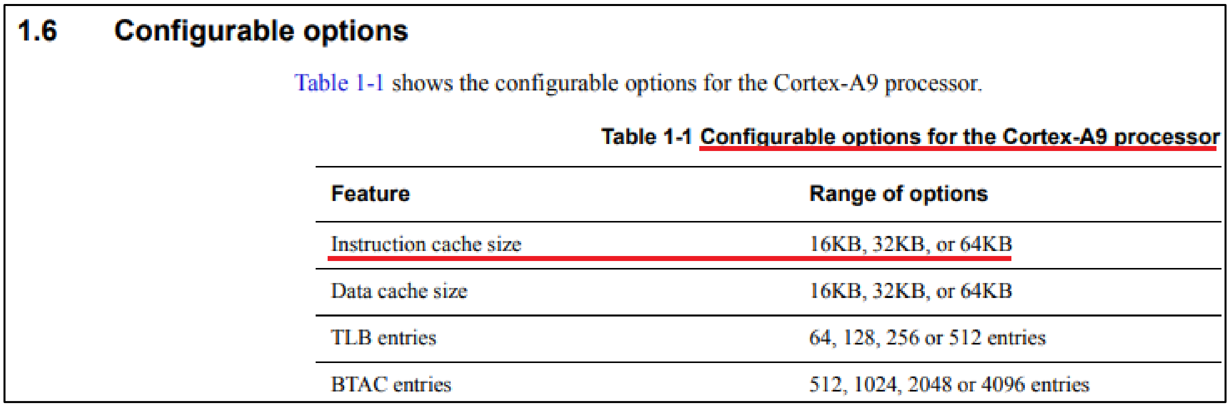
19. For example, the ARM Cortex-A9 in the Panasonic AIHMX707 is a processor that includes an instruction cache. The instruction cache includes multiple cache lines or blocks.



1.1 About the Cortex-A9 processor

The Cortex-A9 processor is a high-performance, low-power, ARM macrocell with an L1 cache subsystem that provides full virtual memory capabilities. The Cortex-A9 processor implements the ARMv7-A architecture and runs 32-bit ARM instructions, 16-bit and 32-bit Thumb instructions, and 8-bit Java bytecodes in Jazelle state.

Source: https://static.docs.arm.com/ddi0388/i/DDI0388I_cortex_a9_r4p1_trm.pdf (Page 13)



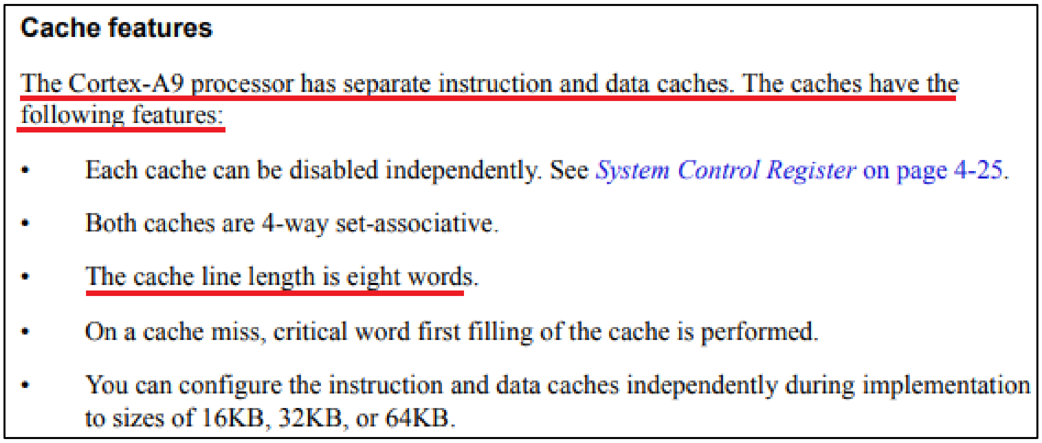
1.6 Configurable options

Table 1-1 shows the configurable options for the Cortex-A9 processor.

Table 1-1 Configurable options for the Cortex-A9 processor

Feature	Range of options
Instruction cache size	16KB, 32KB, or 64KB
Data cache size	16KB, 32KB, or 64KB
TLB entries	64, 128, 256 or 512 entries
BTAC entries	512, 1024, 2048 or 4096 entries

Source: https://static.docs.arm.com/ddi0388/i/DDI0388I_cortex_a9_r4p1_trm.pdf (Page 19)

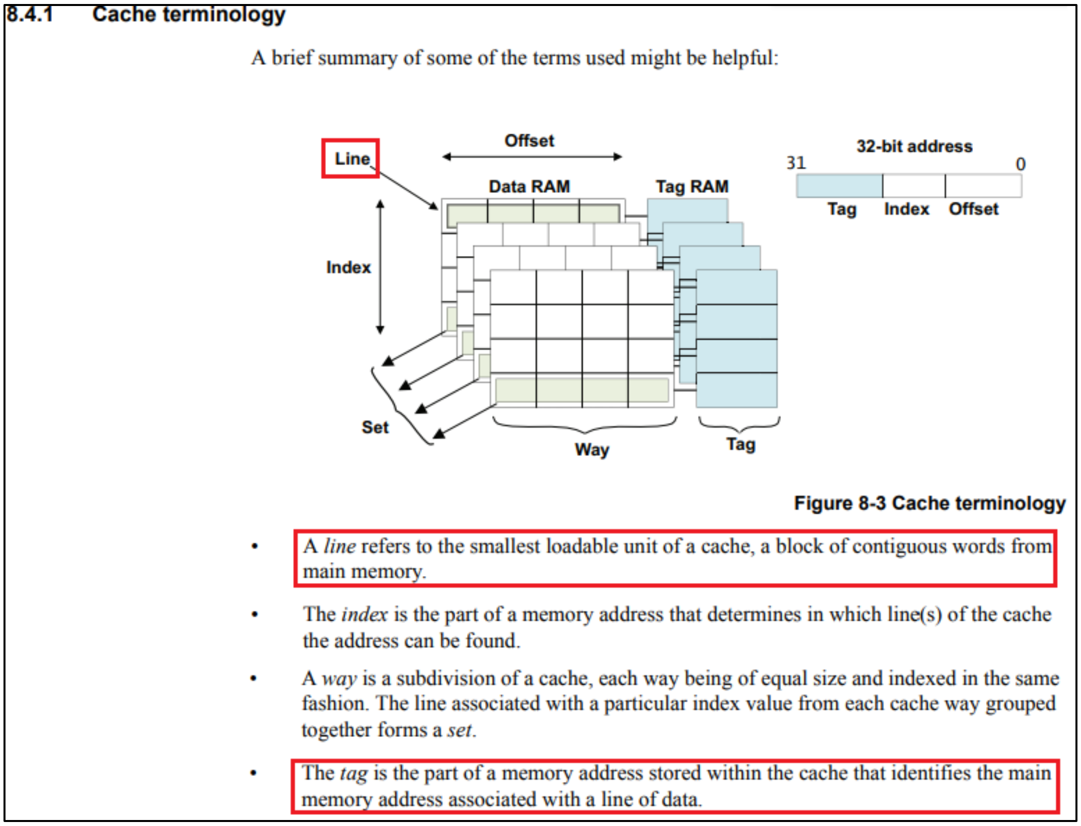


Cache features

The Cortex-A9 processor has separate instruction and data caches. The caches have the following features:

- Each cache can be disabled independently. See *System Control Register* on page 4-25.
- Both caches are 4-way set-associative.
- The cache line length is eight words.
- On a cache miss, critical word first filling of the cache is performed.
- You can configure the instruction and data caches independently during implementation to sizes of 16KB, 32KB, or 64KB.

Source: https://static.docs.arm.com/ddi0388/i/DDI0388I_cortex_a9_r4p1_trm.pdf (Page 113)



8.4.1 Cache terminology

A brief summary of some of the terms used might be helpful:

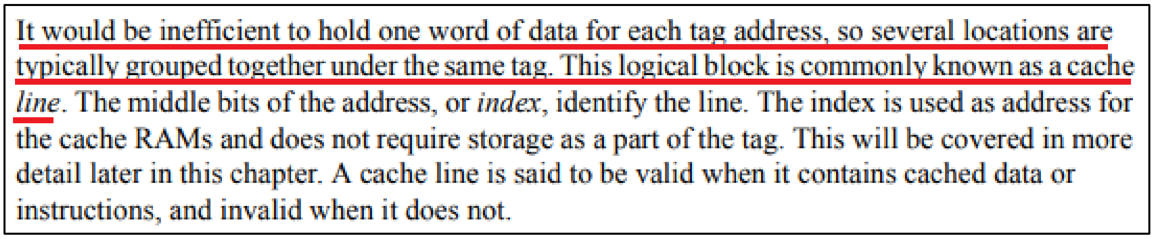
Figure 8-3 Cache terminology

- A *line* refers to the smallest loadable unit of a cache, a block of contiguous words from main memory.
- The *index* is the part of a memory address that determines in which line(s) of the cache the address can be found.
- A *way* is a subdivision of a cache, each way being of equal size and indexed in the same fashion. The line associated with a particular index value from each cache way grouped together forms a *set*.
- The *tag* is the part of a memory address stored within the cache that identifies the main memory address associated with a line of data.

Source: https://documentation-service.arm.com/static/5e909fb6c8052b16087625aa?token=

(Pages 112-113).

Citation 5:



It would be inefficient to hold one word of data for each tag address, so several locations are typically grouped together under the same tag. This logical block is commonly known as a cache *line*. The middle bits of the address, or *index*, identify the line. The index is used as address for the cache RAMs and does not require storage as a part of the tag. This will be covered in more detail later in this chapter. A cache line is said to be valid when it contains cached data or instructions, and invalid when it does not.

Source: https://documentation-service.arm.com/static/5e909fb6c8052b16087625aa?token=

(Page 112).

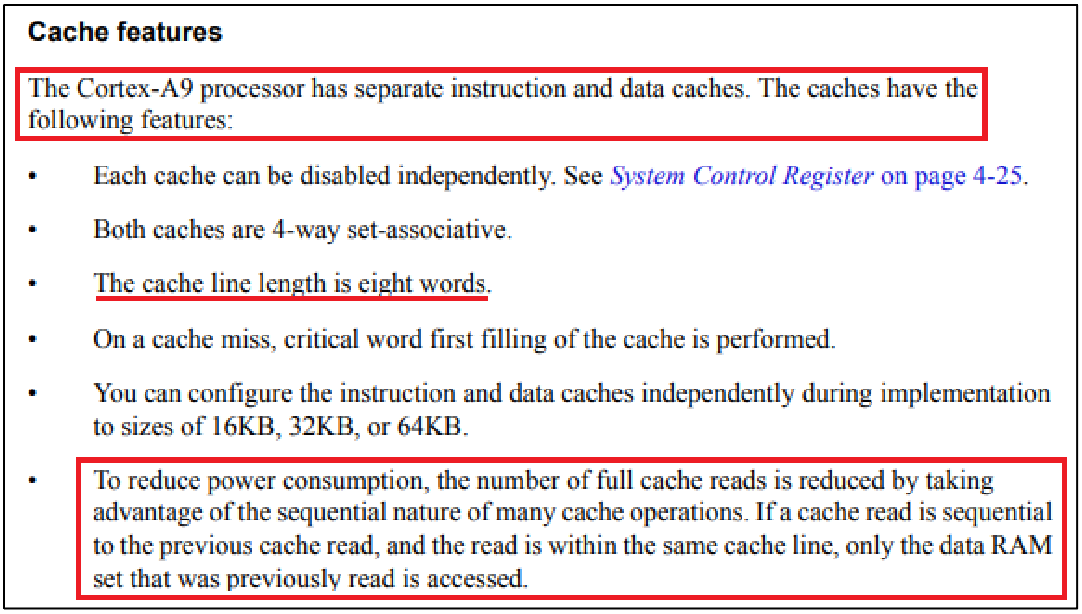
20. The ARM Cortex-A9 processor in the Panasonic AIHMX707 includes a circuit that is configured to generate a power reduction signal. The power reduction signal indicates if a

subsequent instruction to be fetched is in a same block (of a plurality of blocks) as a previous instruction fetched from the instruction cache.

21. For example, the ARM Cortex-A9 processor supports a power reduction method that is operational when an instruction is being accessed from the instruction cache. The instruction cache includes multiple cache lines or blocks, and each cache line or block is associated with a tag value. These tag values are stored in a tag RAM. The cache also includes data RAM for storing the instructions.

22. If a sequential (or subsequent) instruction to be read from the instruction cache is in the same cache line or block as the previous instruction, only the data RAM of the cache is accessed for the instruction, and the tag RAM is *not* accessed because the sequential instruction resides in the same cache line or block.

23. Accordingly, the processor includes a circuit that sends a signal ("power reduction signal") if a sequential instruction to be accessed from the instruction cache is identified as being in the same cache line or block.

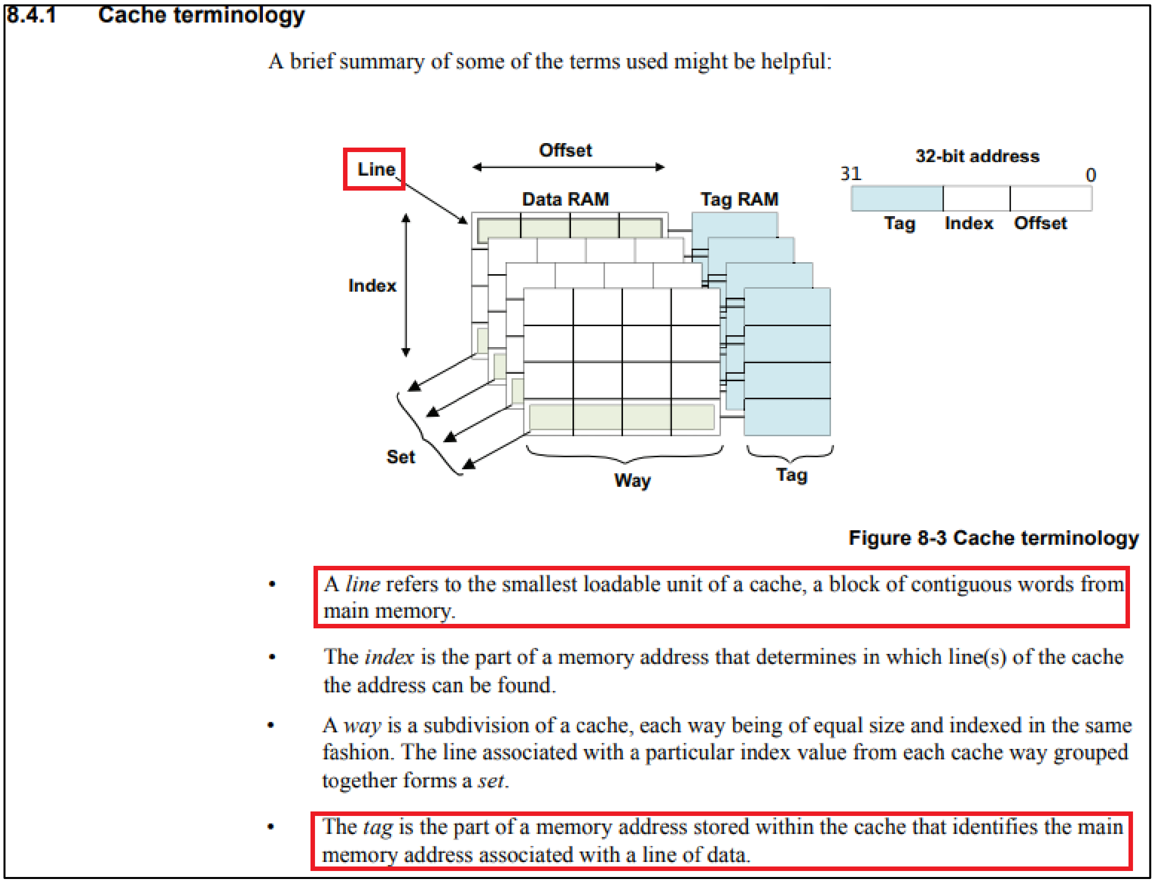


Cache features

The Cortex-A9 processor has separate instruction and data caches. The caches have the following features:

- Each cache can be disabled independently. See *System Control Register* on page 4-25.
- Both caches are 4-way set-associative.
- The cache line length is eight words.
- On a cache miss, critical word first filling of the cache is performed.
- You can configure the instruction and data caches independently during implementation to sizes of 16KB, 32KB, or 64KB.
- To reduce power consumption, the number of full cache reads is reduced by taking advantage of the sequential nature of many cache operations. If a cache read is sequential to the previous cache read, and the read is within the same cache line, only the data RAM set that was previously read is accessed.

Source: https://static.docs.arm.com/ddi0388/i/DDI0388I_cortex_a9_r4p1_trm.pdf (Page 113)



8.4.1 Cache terminology

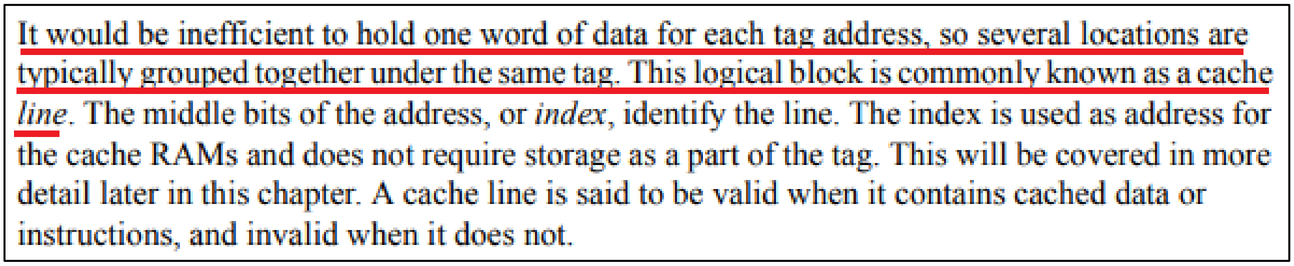
A brief summary of some of the terms used might be helpful:

Figure 8-3 Cache terminology

- A *line* refers to the smallest loadable unit of a cache, a block of contiguous words from main memory.
- The *index* is the part of a memory address that determines in which line(s) of the cache the address can be found.
- A *way* is a subdivision of a cache, each way being of equal size and indexed in the same fashion. The line associated with a particular index value from each cache way grouped together forms a *set*.
- The *tag* is the part of a memory address stored within the cache that identifies the main memory address associated with a line of data.

Source: https://documentation-service.arm.com/static/5e909fb6c8052b16087625aa?token=

(Pages 112-113).



It would be inefficient to hold one word of data for each tag address, so several locations are typically grouped together under the same tag. This logical block is commonly known as a cache *line*. The middle bits of the address, or *index*, identify the line. The index is used as address for the cache RAMs and does not require storage as a part of the tag. This will be covered in more detail later in this chapter. A cache line is said to be valid when it contains cached data or instructions, and invalid when it does not.

Source: https://documentation-service.arm.com/static/5e909fb6c8052b16087625aa?token=

(Page 112).

24. Panasonic has had knowledge of the ’959 Patent at least as of the date when it was notified of the filing of this action.

25. Liberty Patents has been damaged as a result of the infringing conduct by Panasonic alleged above. Thus, Panasonic 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.

26. 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 ’959 Patent.

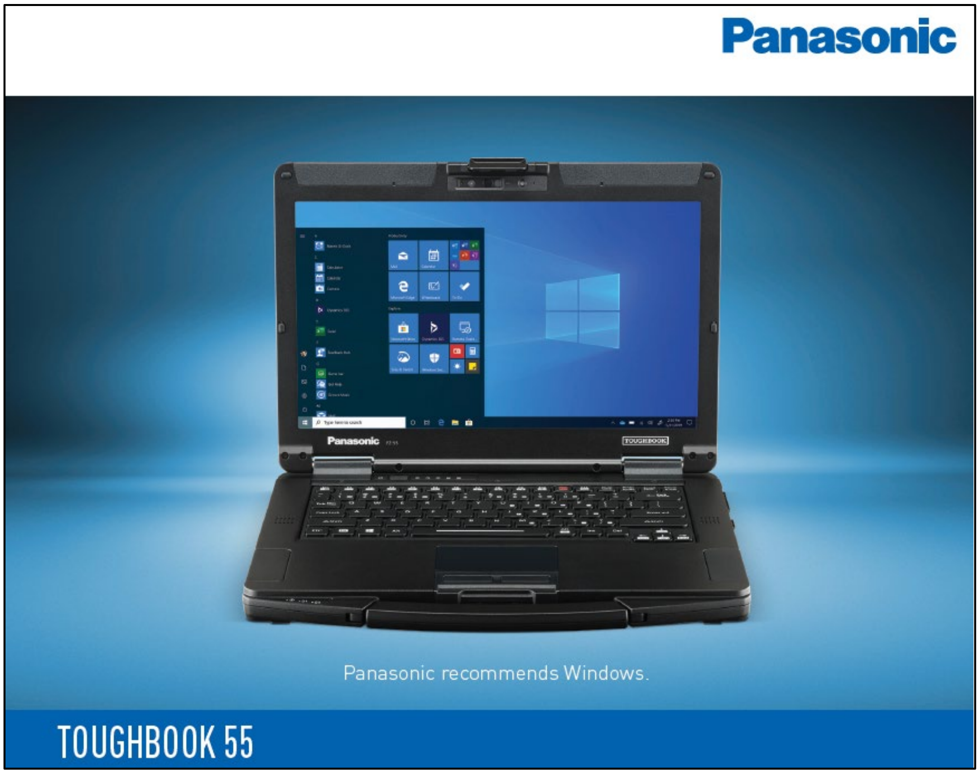
COUNT II

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

27. 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.”

28. 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.

29. Panasonic made, had made, used, imported, provided, supplied, distributed, sold, and/or offered for sale products and/or systems including, for example, its Panasonic Toughbook 55 and other products (e.g., Panasonic Toughbooks CF-20, CF-33, CF-54 Prime, FZ-55, etc.) including the “Always On Charging” feature (“accused products”):



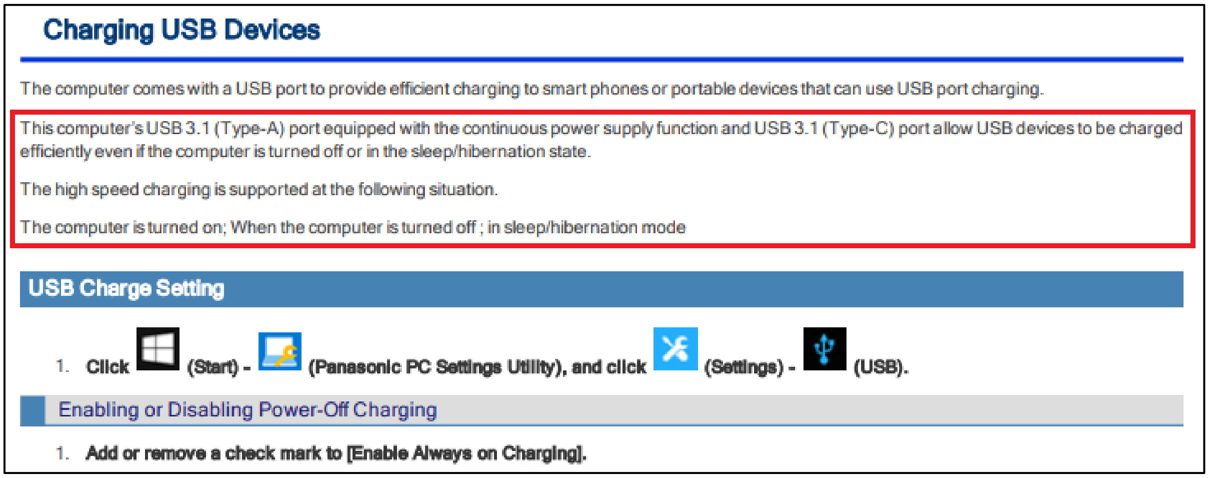
Panasonic

Panasonic recommends Windows.

TOUGHBOOK 55

Source: https://na.panasonic.com/ns/271751_TOUGHBOOK_55_spec_sheet_january_2020.pdf

(Page 1)



Charging USB Devices

The computer comes with a USB port to provide efficient charging to smart phones or portable devices that can use USB port charging.

This computer's USB 3.1 (Type-A) port equipped with the continuous power supply function and USB 3.1 (Type-C) port allow USB devices to be charged efficiently even if the computer is turned off or in the sleep/hibernation state.

The high speed charging is supported at the following situation.

The computer is turned on; When the computer is turned off ; in sleep/hibernation mode

USB Charge Setting

1. **Click (Start) - (Panasonic PC Settings Utility), and click (Settings) - (USB).**

Enabling or Disabling Power-Off Charging

1. **Add or remove a check mark to [Enable Always on Charging].**

Source: https://na.panasonic.com/ns/271398_TOUGHBOOK_55_Reference_Manual_11-19.pdf

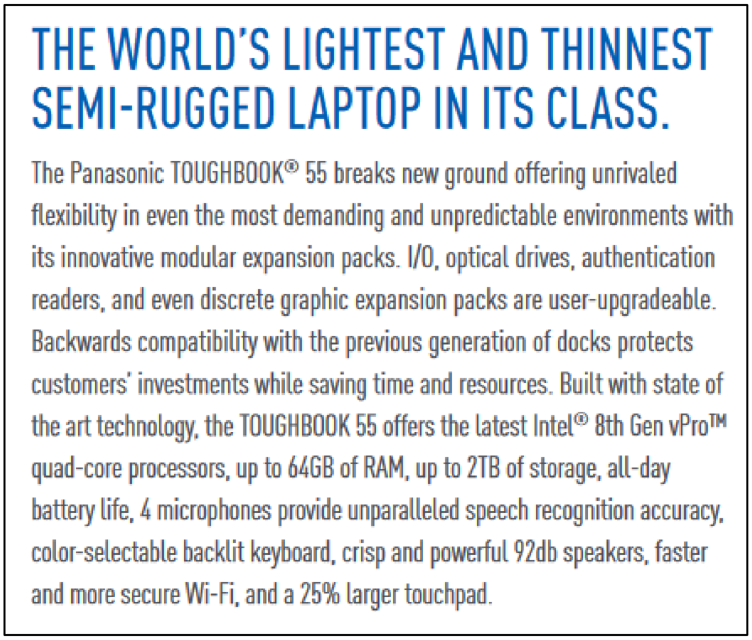
(Page 109)

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

31. The Panasonic Toughbook 55 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

32. For example, the Panasonic Toughbook 55 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 Toughbook’s Power On mode, Sleep mode and Shut Down mode, respectively. The processor functions differently depending on the current operating mode.

33. 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.

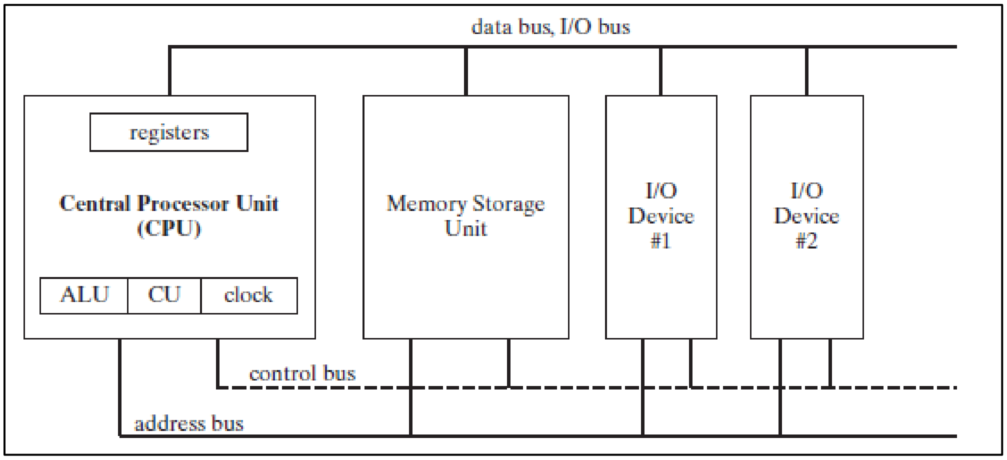


THE WORLD'S LIGHTEST AND THINNEST SEMI-RUGGED LAPTOP IN ITS CLASS.

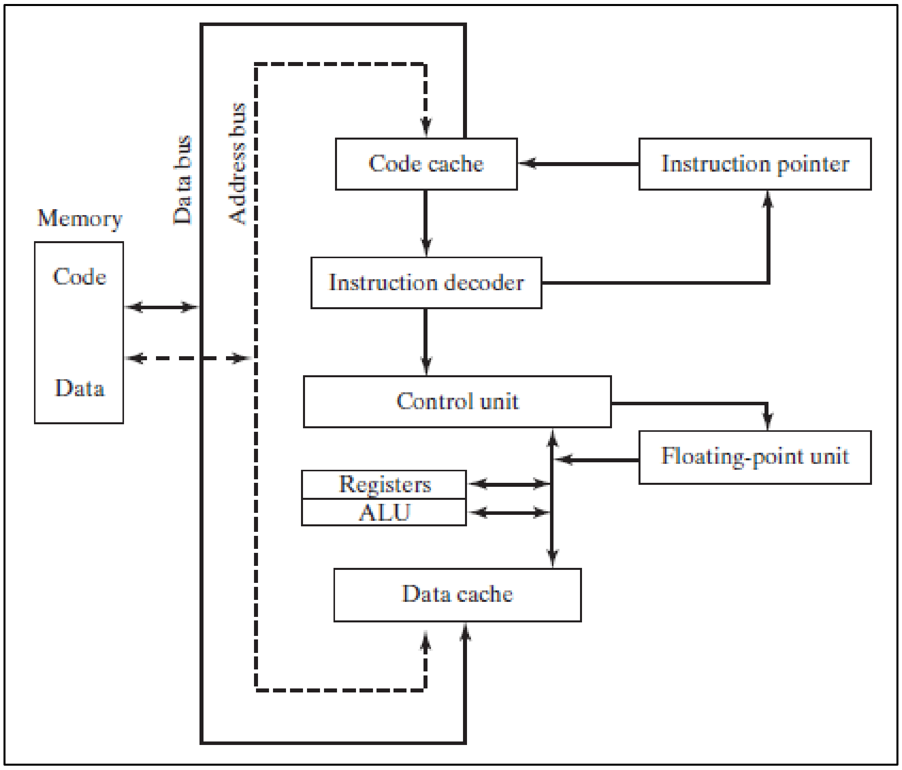
The Panasonic TOUGHBOOK® 55 breaks new ground offering unrivaled flexibility in even the most demanding and unpredictable environments with its innovative modular expansion packs. I/O, optical drives, authentication readers, and even discrete graphic expansion packs are user-upgradeable. Backwards compatibility with the previous generation of docks protects customers' investments while saving time and resources. Built with state of the art technology, the TOUGHBOOK 55 offers the latest Intel® 8th Gen vPro™ quad-core processors, up to 64GB of RAM, up to 2TB of storage, all-day battery life, 4 microphones provide unparalleled speech recognition accuracy, color-selectable backlit keyboard, crisp and powerful 92db speakers, faster and more secure Wi-Fi, and a 25% larger touchpad.

Source: https://na.panasonic.com/ns/271751_TOUGHBOOK_55_spec_sheet_january_2020.pdf

(Page 1)

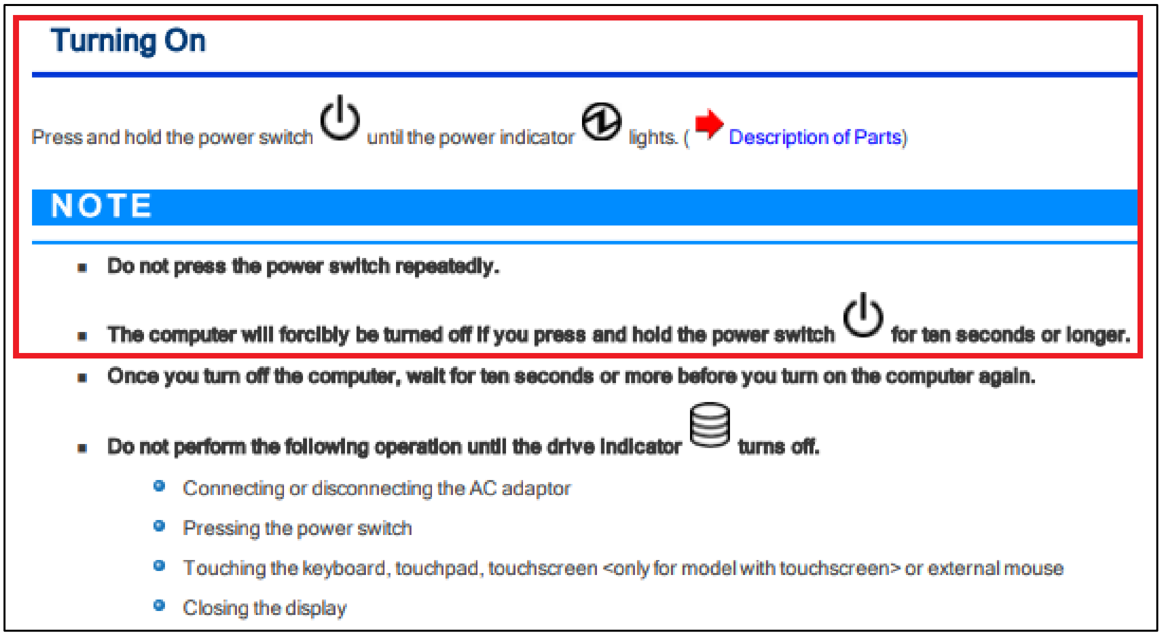


Source: https://www.allaboutcircuits.com/technical-articles/an-introduction-to-x86-processor-architecture/



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

34. The following citations disclose different operating modes of the Panasonic Toughbook 55, including Shut Down mode, Sleep mode, and Power On mode. The computer operates differently according to the current operating mode.



Turning On

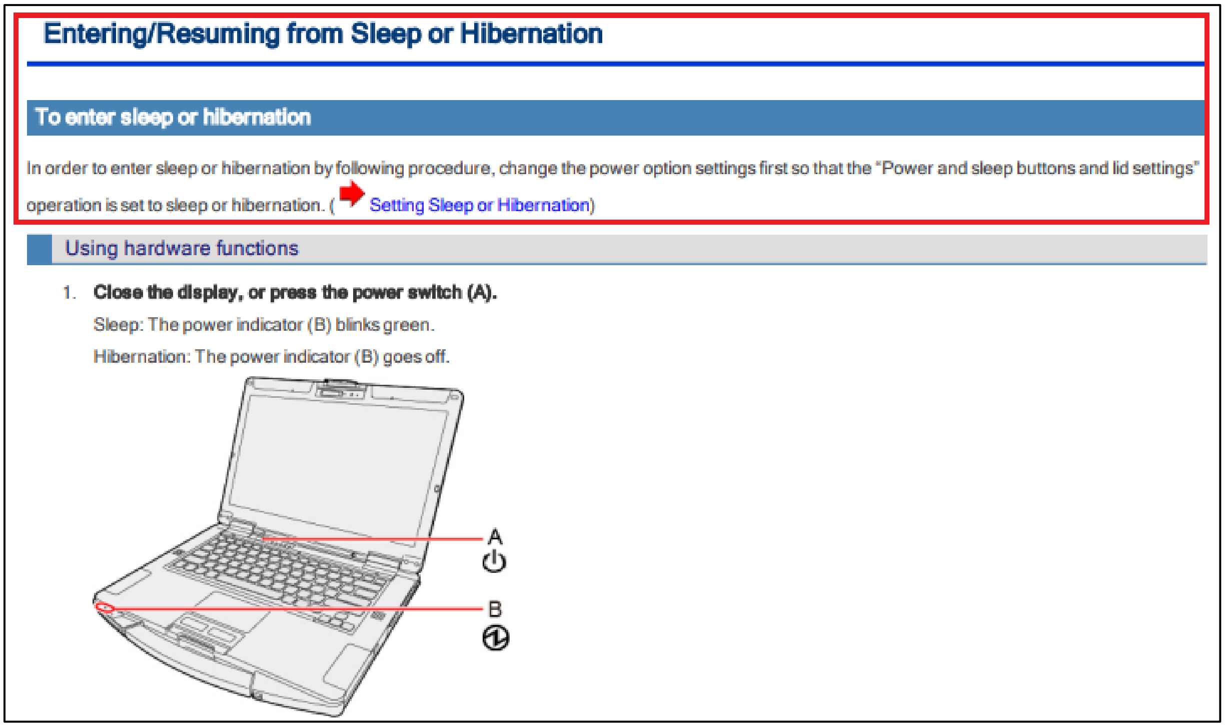
Press and hold the power switch ⏻ until the power indicator ⏼ lights. (➡ Description of Parts)

NOTE

- **Do not press the power switch repeatedly.**
- **The computer will forcibly be turned off if you press and hold the power switch ⏻ for ten seconds or longer.**
- **Once you turn off the computer, wait for ten seconds or more before you turn on the computer again.**
- **Do not perform the following operation until the drive indicator ⛁ turns off.**
  - Connecting or disconnecting the AC adaptor
  - Pressing the power switch
  - Touching the keyboard, touchpad, touchscreen <only for model with touchscreen> or external mouse
  - Closing the display

Source: https://na.panasonic.com/ns/271398_TOUGHBOOK_55_Reference_Manual_11-19.pdf

(Page 24)



Entering/Resuming from Sleep or Hibernation

To enter sleep or hibernation

In order to enter sleep or hibernation by following procedure, change the power option settings first so that the "Power and sleep buttons and lid settings" operation is set to sleep or hibernation. (➡ Setting Sleep or Hibernation)

Using hardware functions

1. **Close the display, or press the power switch (A).**

   Sleep: The power indicator (B) blinks green.

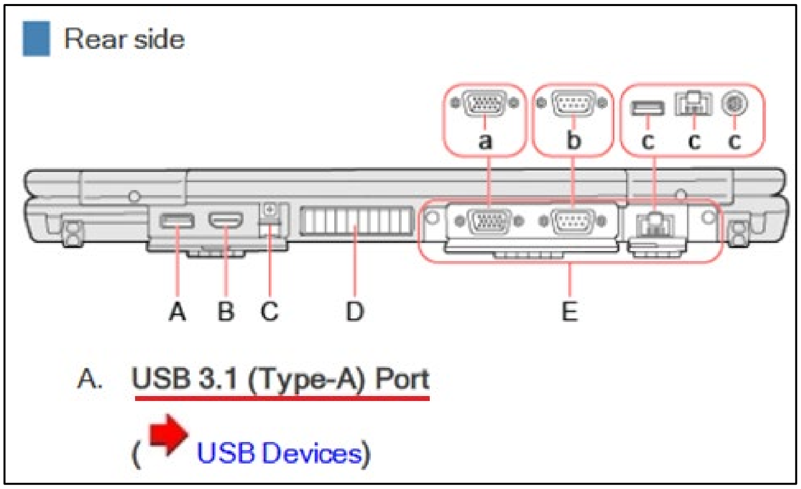
   Hibernation: The power indicator (B) goes off.

Source: https://na.panasonic.com/ns/271398_TOUGHBOOK_55_Reference_Manual_11-19.pdf

(Page 48)

35. The Panasonic Toughbook 55 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.

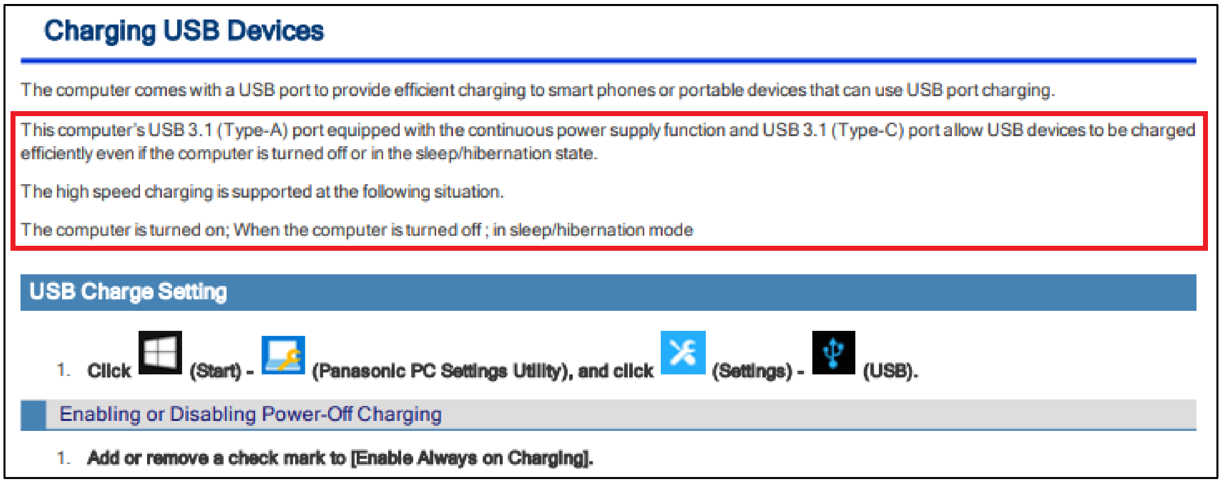
36. For example, Panasonic Toughbook 55 has an "Always On Charging" feature that allows a user to charge ("second operating function") USB connected devices (such as such as a mobile devices, cameras, activity trackers, smartwatches, etc.) even when the device is in the Shut Down or Off mode. Mobile devices can be charged using the designated USB port having the "Always On Charging" feature without requiring the Panasonic Toughbook 55 to be in 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.


Rear side

A. USB 3.1 (Type-A) Port

(USB Devices)

Source: https://na.panasonic.com/ns/271398_TOUGHBOOK_55_Reference_Manual_11-19.pdf (Page 21)



Charging USB Devices

The computer comes with a USB port to provide efficient charging to smart phones or portable devices that can use USB port charging.

This computer's USB 3.1 (Type-A) port equipped with the continuous power supply function and USB 3.1 (Type-C) port allow USB devices to be charged efficiently even if the computer is turned off or in the sleep/hibernation state.

The high speed charging is supported at the following situation.

The computer is turned on; When the computer is turned off ; in sleep/hibernation mode

USB Charge Setting

1. **Click (Start) - (Panasonic PC Settings Utility), and click (Settings) - (USB).**

Enabling or Disabling Power-Off Charging

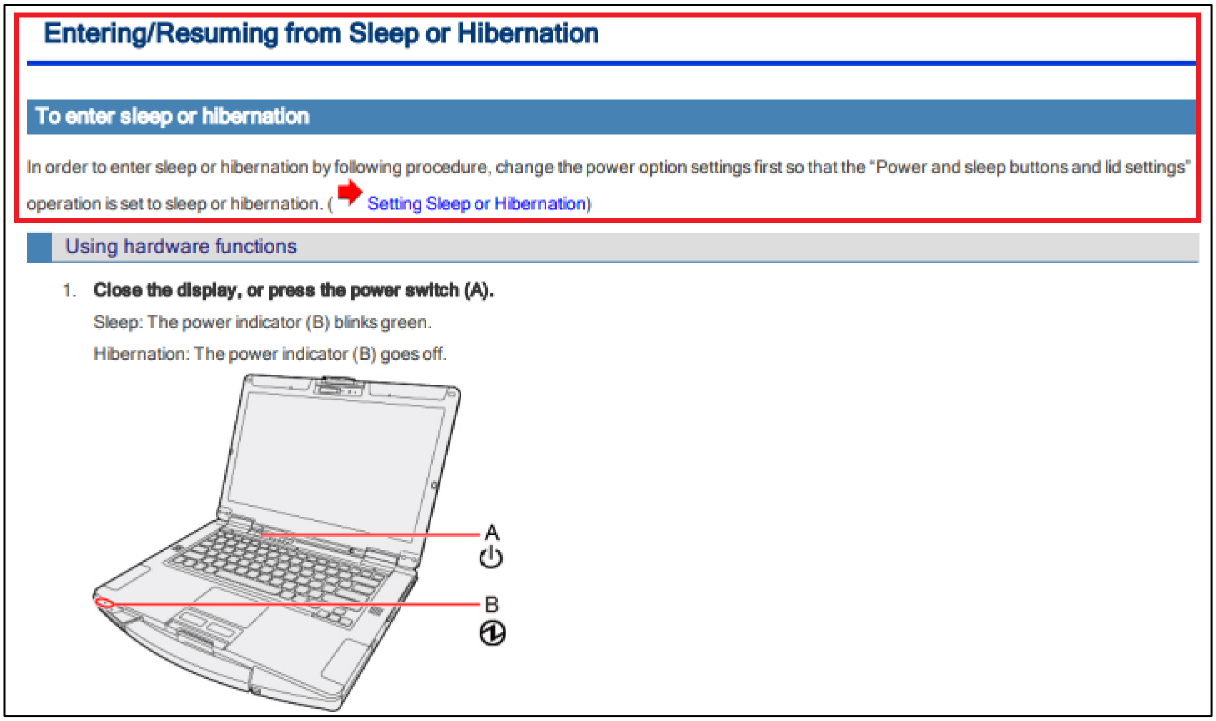
1. **Add or remove a check mark to [Enable Always on Charging].**

Source: https://na.panasonic.com/ns/271398_TOUGHBOOK_55_Reference_Manual_11-19.pdf (Page 109)

37. The Panasonic Toughbook 55 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.

38. For example, the Panasonic Toughbook 55 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 Panasonic Toughbook 55 includes corresponding circuitry ("third group of circuitry") that activates and deactivates the Sleep mode.

39. 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").



Entering/Resuming from Sleep or Hibernation

To enter sleep or hibernation

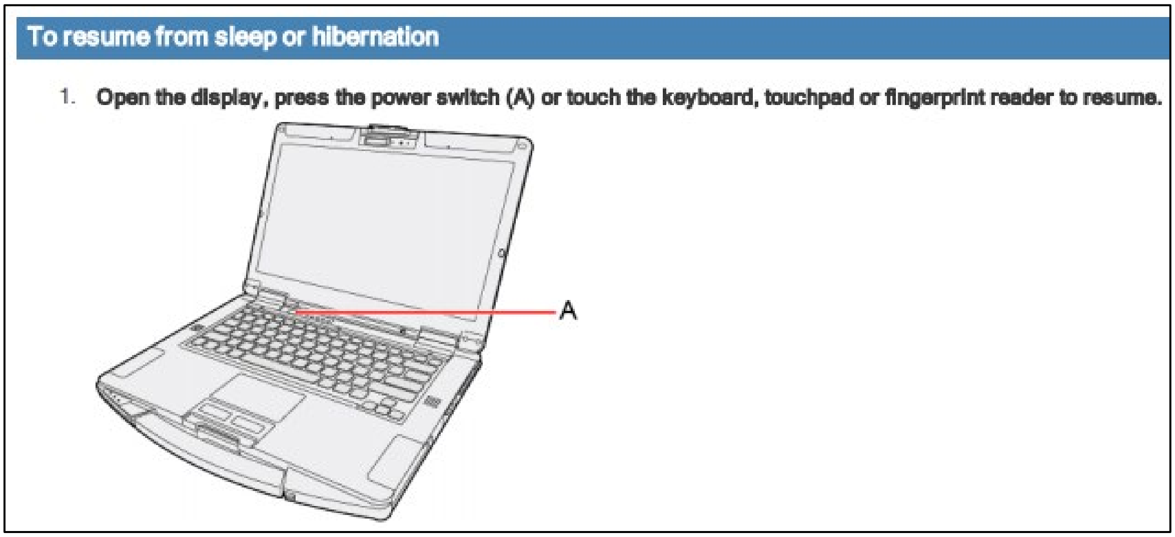
In order to enter sleep or hibernation by following procedure, change the power option settings first so that the "Power and sleep buttons and lid settings" operation is set to sleep or hibernation. (➡ Setting Sleep or Hibernation)

Using hardware functions

1. **Close the display, or press the power switch (A).**
   Sleep: The power indicator (B) blinks green.
   Hibernation: The power indicator (B) goes off.

Source: https://na.panasonic.com/ns/271398_TOUGHBOOK_55_Reference_Manual_11-19.pdf

(Page 48)

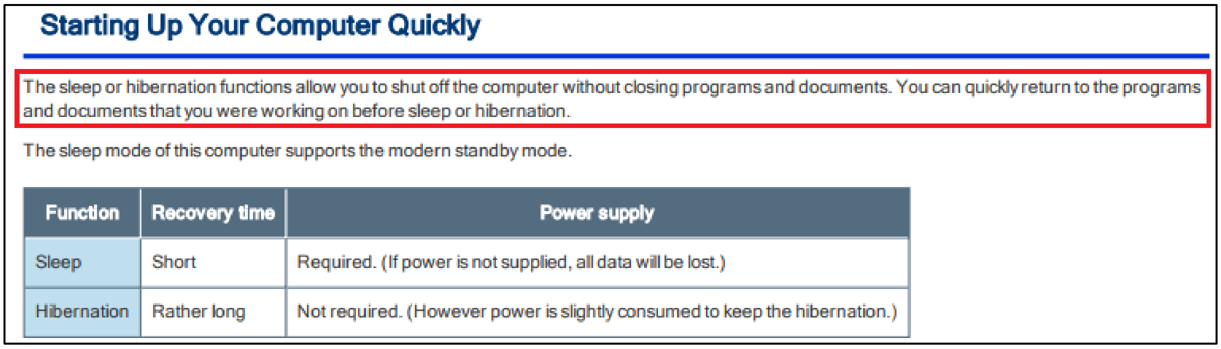


To resume from sleep or hibernation

1. **Open the display, press the power switch (A) or touch the keyboard, touchpad or fingerprint reader to resume.**

Source: https://na.panasonic.com/ns/271398_TOUGHBOOK_55_Reference_Manual_11-19.pdf

(Page 49)



Starting Up Your Computer Quickly

The sleep or hibernation functions allow you to shut off the computer without closing programs and documents. You can quickly return to the programs and documents that you were working on before sleep or hibernation.

The sleep mode of this computer supports the modern standby mode.

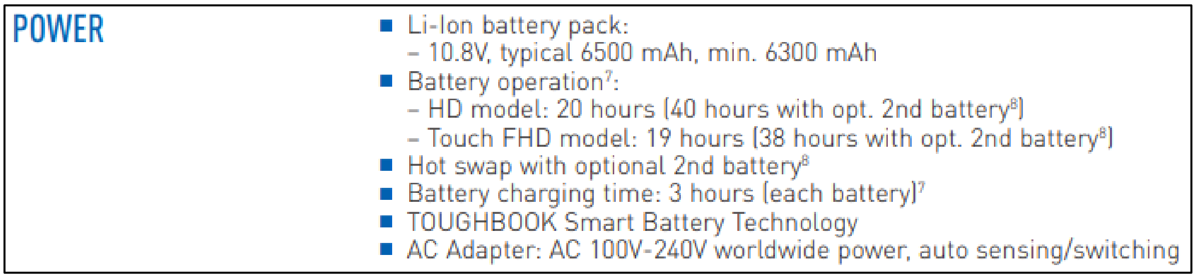
Function	Recovery time	Power supply
Sleep	Short	Required. (If power is not supplied, all data will be lost.)
Hibernation	Rather long	Not required. (However power is slightly consumed to keep the hibernation.)

Source: https://na.panasonic.com/ns/271398_TOUGHBOOK_55_Reference_Manual_11-19.pdf

(Page 47)

40. The Panasonic Toughbook 55 includes power providing means for providing power to the first group of circuitry, the second group of circuitry, and the third group of circuitry.

41. For example, the Panasonic Toughbook 55 includes a Lithium ion 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).

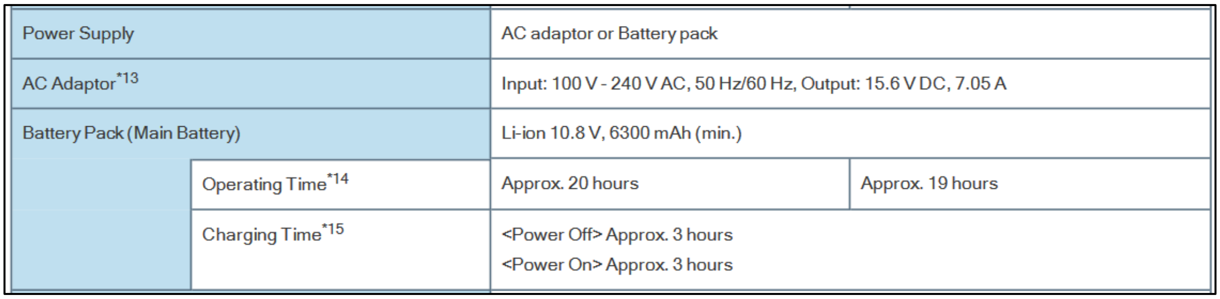


POWER

- Li-Ion battery pack:
  - 10.8V, typical 6500 mAh, min. 6300 mAh
- Battery operation[7]:
  - HD model: 20 hours (40 hours with opt. 2nd battery[8])
  - Touch FHD model: 19 hours (38 hours with opt. 2nd battery[8])
- Hot swap with optional 2nd battery[8]
- Battery charging time: 3 hours (each battery)[7]
- TOUGHBOOK Smart Battery Technology
- AC Adapter: AC 100V-240V worldwide power, auto sensing/switching

Source: https://na.panasonic.com/ns/271751_TOUGHBOOK_55_spec_sheet_january_2020.pdf

(Page 2)



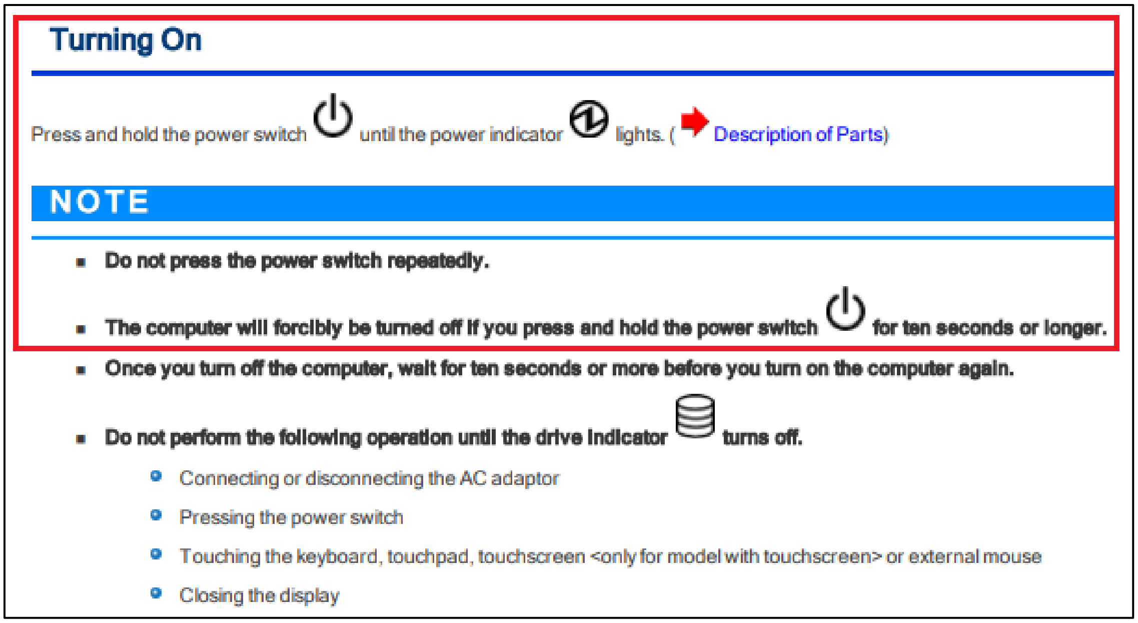
Power Supply		AC adaptor or Battery pack	
AC Adaptor[*13]		Input: 100 V - 240 V AC, 50 Hz/60 Hz, Output: 15.6 V DC, 7.05 A	
Battery Pack (Main Battery)		Li-ion 10.8 V, 6300 mAh (min.)	
	Operating Time[*14]	Approx. 20 hours	Approx. 19 hours
	Charging Time[*15]	<Power Off> Approx. 3 hours <Power On> Approx. 3 hours	

Source: https://na.panasonic.com/ns/271398_TOUGHBOOK_55_Reference_Manual_11-19.pdf (Page 157)

42. The Panasonic Toughbook 55 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.

43. For example, the Panasonic Toughbook 55 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 "Always On Charging" feature enables charging of a mobile device through the designated USB port during Shut Down mode ("second operating function").

44. The processor of the Panasonic Toughbook 55 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.



Turning On

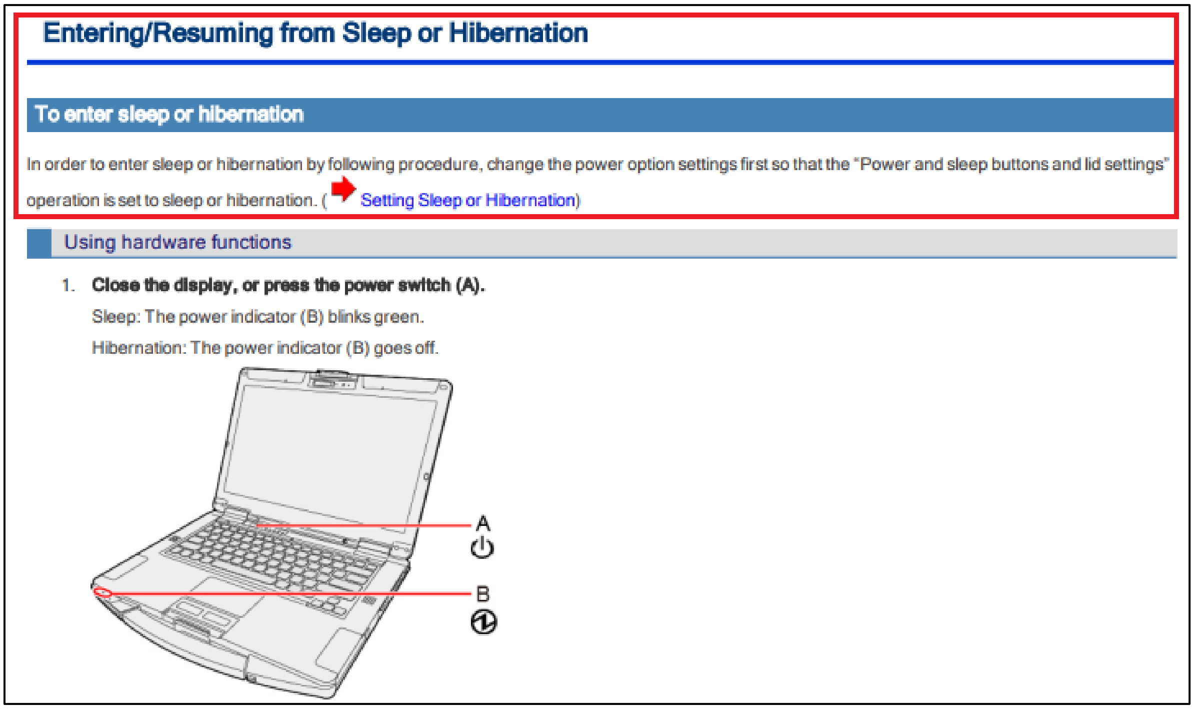
Press and hold the power switch ⏻ until the power indicator ⏻ lights. (➡ Description of Parts)

NOTE

- **Do not press the power switch repeatedly.**
- **The computer will forcibly be turned off if you press and hold the power switch ⏻ for ten seconds or longer.**
- **Once you turn off the computer, wait for ten seconds or more before you turn on the computer again.**
- **Do not perform the following operation until the drive indicator ⛁ turns off.**
  - Connecting or disconnecting the AC adaptor
  - Pressing the power switch
  - Touching the keyboard, touchpad, touchscreen <only for model with touchscreen> or external mouse
  - Closing the display

Source: https://na.panasonic.com/ns/271398_TOUGHBOOK_55_Reference_Manual_11-19.pdf

(Page 24)



Entering/Resuming from Sleep or Hibernation

To enter sleep or hibernation

In order to enter sleep or hibernation by following procedure, change the power option settings first so that the "Power and sleep buttons and lid settings" operation is set to sleep or hibernation. (➡ Setting Sleep or Hibernation)

Using hardware functions

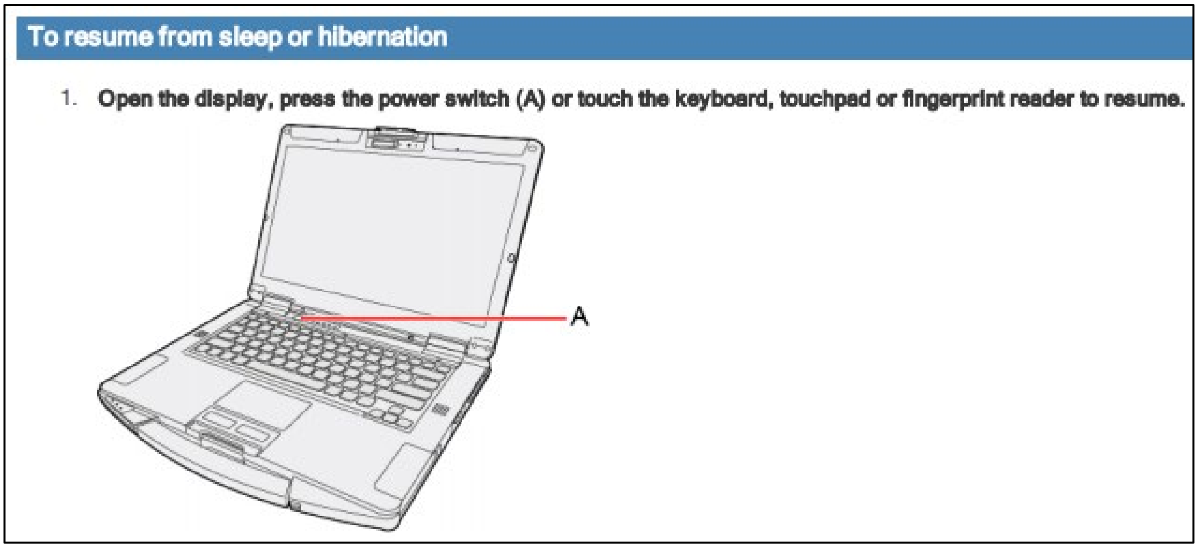
1. **Close the display, or press the power switch (A).**

   Sleep: The power indicator (B) blinks green.

   Hibernation: The power indicator (B) goes off.

Source: https://na.panasonic.com/ns/271398_TOUGHBOOK_55_Reference_Manual_11-19.pdf

(Page 48)

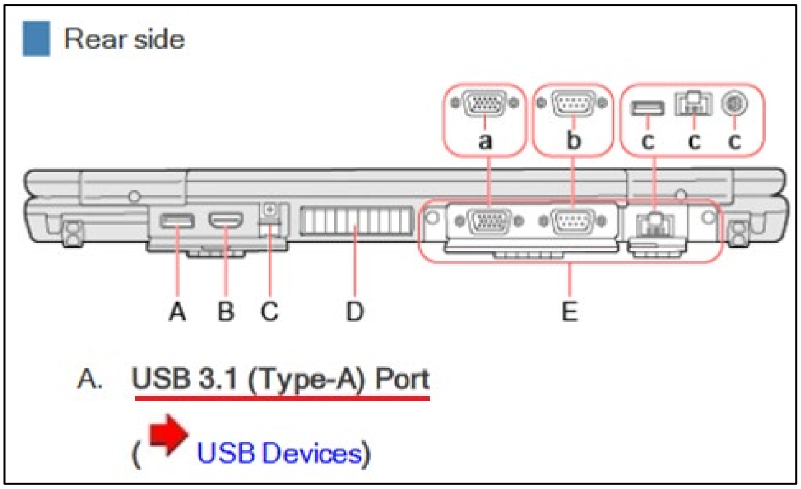


To resume from sleep or hibernation

1. Open the display, press the power switch (A) or touch the keyboard, touchpad or fingerprint reader to resume.

Source: https://na.panasonic.com/ns/271398_TOUGHBOOK_55_Reference_Manual_11-19.pdf

(Page 49)



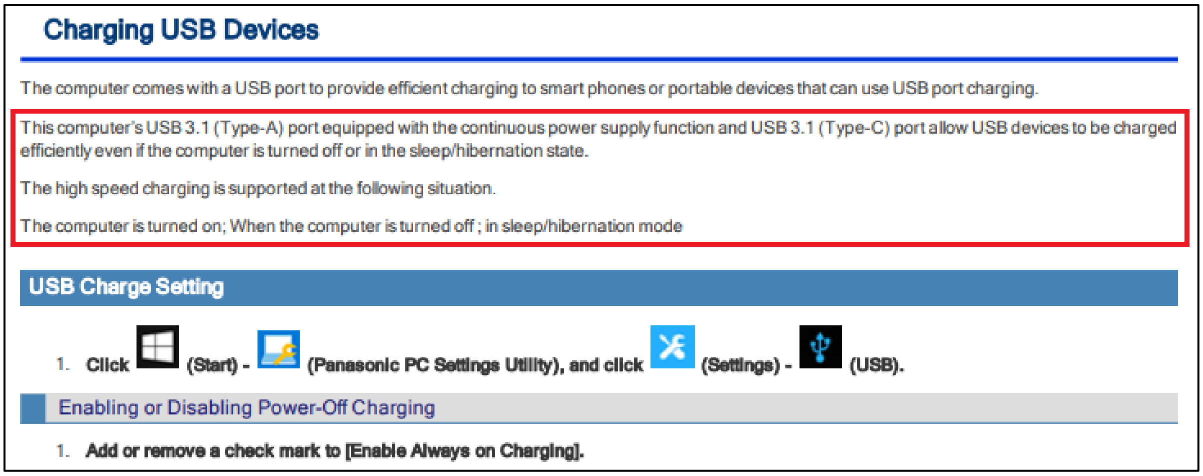
Rear side

A. USB 3.1 (Type-A) Port

(USB Devices)

Source: https://na.panasonic.com/ns/271398_TOUGHBOOK_55_Reference_Manual_11-19.pdf

(Page 21)



Charging USB Devices

The computer comes with a USB port to provide efficient charging to smart phones or portable devices that can use USB port charging.

This computer's USB 3.1 (Type-A) port equipped with the continuous power supply function and USB 3.1 (Type-C) port allow USB devices to be charged efficiently even if the computer is turned off or in the sleep/hibernation state.

The high speed charging is supported at the following situation.

The computer is turned on; When the computer is turned off ; in sleep/hibernation mode

USB Charge Setting

1. Click (Start) - (Panasonic PC Settings Utility), and click (Settings) - (USB).

Enabling or Disabling Power-Off Charging

1. Add or remove a check mark to [Enable Always on Charging].

Source: https://na.panasonic.com/ns/271398_TOUGHBOOK_55_Reference_Manual_11-19.pdf

(Page 109)

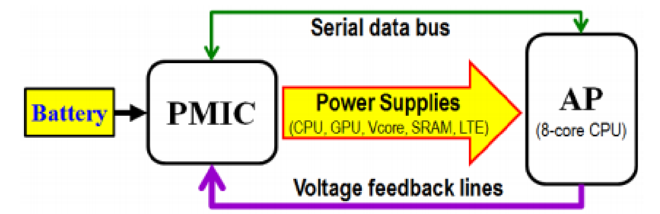


Fig. 1. Common interconnection between PMIC and AP in mobile platforms.

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

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

46. Liberty Patents has been damaged as a result of Panasonic's infringing conduct alleged above. Thus, Panasonic 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.

47. 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.

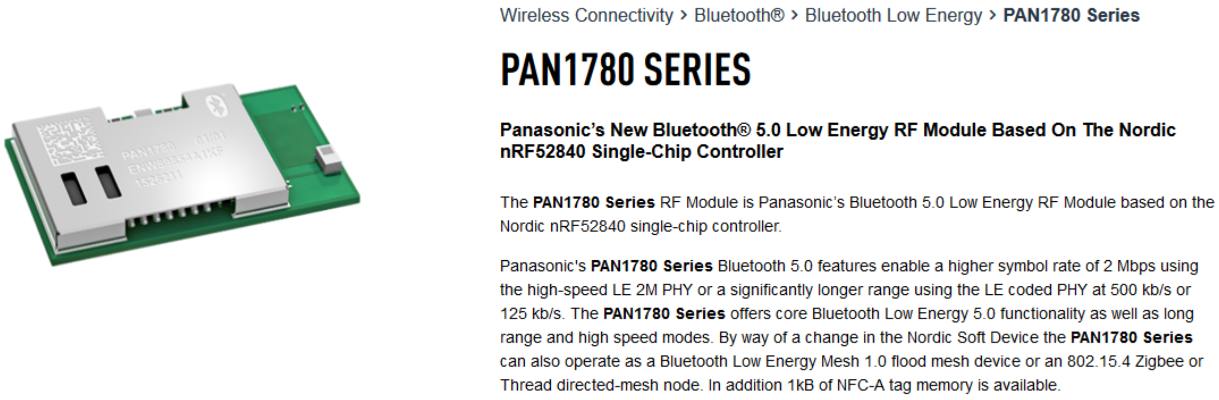
COUNT III

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

48. 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.”

49. 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.

50. Panasonic made, had made, used, imported, provided, supplied, distributed, sold, and/or offered for sale products and/or systems including, for example, PSA Certified products such as the PAN1780 Series Bluetooth® 5.0 Low Energy RF Modules (“accused products”):



Wireless Connectivity › Bluetooth® › Bluetooth Low Energy › **PAN1780 Series**

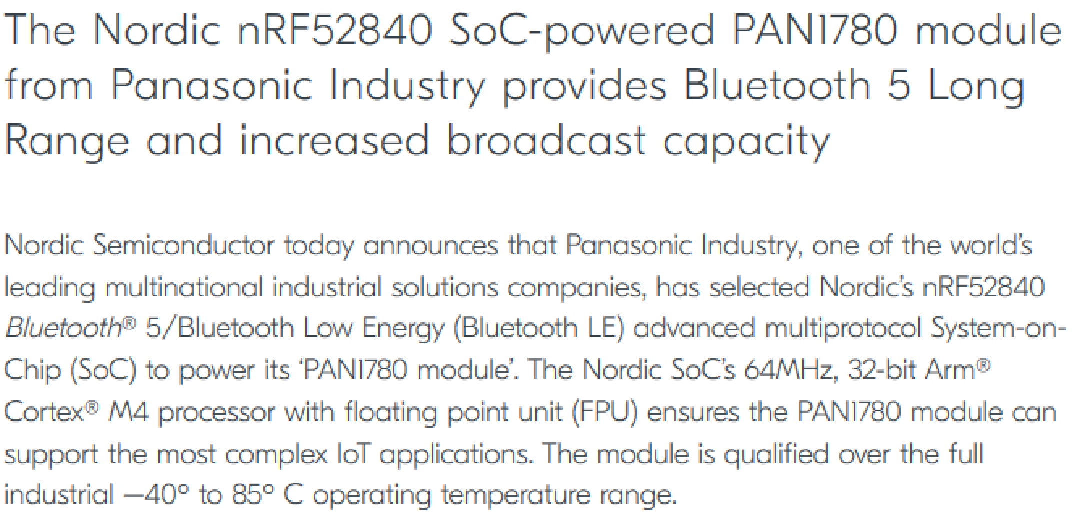
PAN1780 SERIES

Panasonic’s New Bluetooth® 5.0 Low Energy RF Module Based On The Nordic nRF52840 Single-Chip Controller

The **PAN1780 Series** RF Module is Panasonic’s Bluetooth 5.0 Low Energy RF Module based on the Nordic nRF52840 single-chip controller.

Panasonic's **PAN1780 Series** Bluetooth 5.0 features enable a higher symbol rate of 2 Mbps using the high-speed LE 2M PHY or a significantly longer range using the LE coded PHY at 500 kb/s or 125 kb/s. The **PAN1780 Series** offers core Bluetooth Low Energy 5.0 functionality as well as long range and high speed modes. By way of a change in the Nordic Soft Device the **PAN1780 Series** can also operate as a Bluetooth Low Energy Mesh 1.0 flood mesh device or an 802.15.4 Zigbee or Thread directed-mesh node. In addition 1kB of NFC-A tag memory is available.

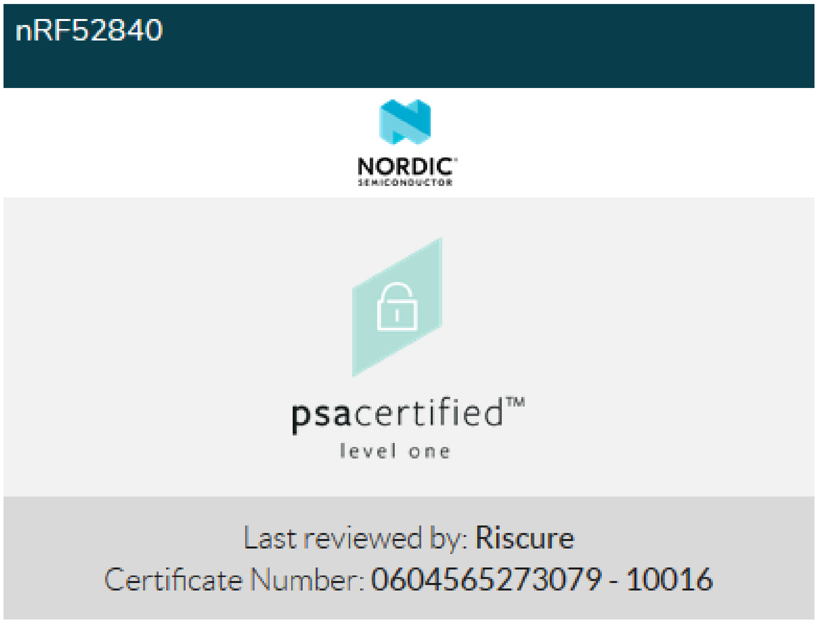
Source: https://na.industrial.panasonic.com/products/wireless-connectivity/bluetooth/lineup/bluetooth-low-energy/series/128560



The Nordic nRF52840 SoC-powered PAN1780 module from Panasonic Industry provides Bluetooth 5 Long Range and increased broadcast capacity

Nordic Semiconductor today announces that Panasonic Industry, one of the world's leading multinational industrial solutions companies, has selected Nordic's nRF52840 *Bluetooth*® 5/Bluetooth Low Energy (Bluetooth LE) advanced multiprotocol System-on-Chip (SoC) to power its 'PAN1780 module'. The Nordic SoC's 64MHz, 32-bit Arm® Cortex® M4 processor with floating point unit (FPU) ensures the PAN1780 module can support the most complex IoT applications. The module is qualified over the full industrial –40° to 85° C operating temperature range.

Source: www.nordicsemi.com/News/2020/01/Bluetooth-module-allows-Panasonic-Industry-customers-to-develop-applications



nRF52840

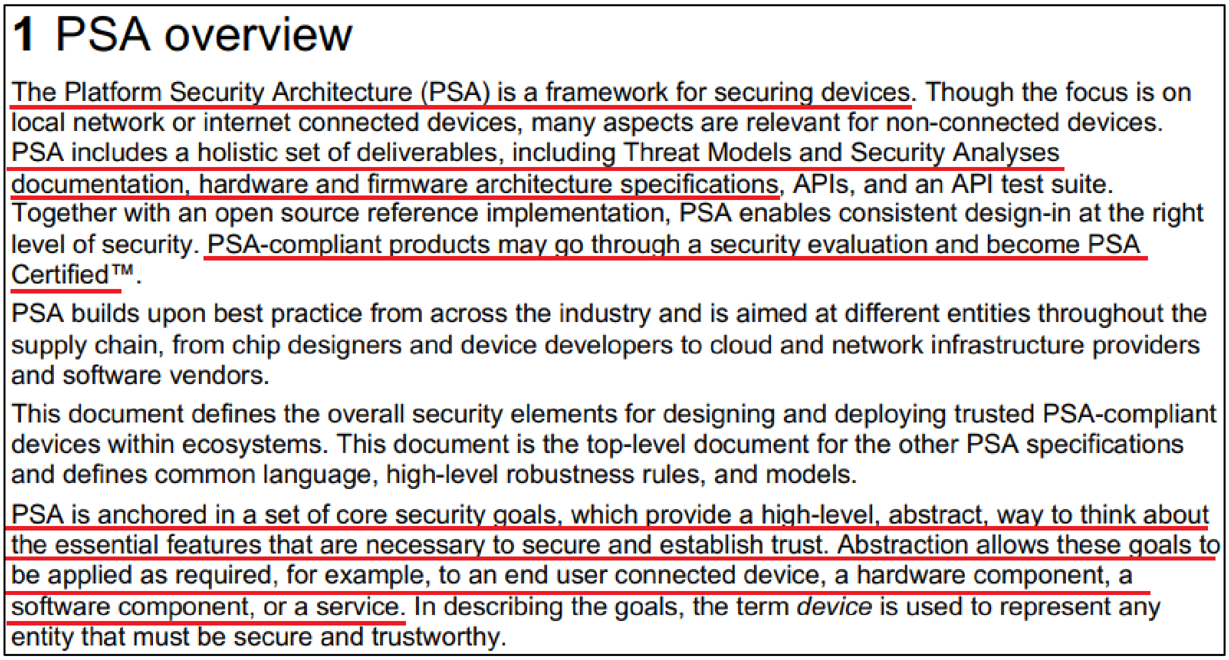
Last reviewed by: Riscure
Certificate Number: 0604565273079 - 10016

Source: www.psacertified.org/certified-products/?_standard=psa-certified-chip-vendor&_type=security-level-1&_company=nordic-semiconductor

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

52. For example, the PSA Certified Panasonic PAN1780 Series Bluetooth® 5.0 Low Energy RF Modules are embedded systems capable of automatically updating system software. PSA (Platform Security Architecture) is a framework designed for providing security features for products such as ICs, chipsets, SOCs, etc. For a product to become PSA certified, it must comply with specific hardware and software requirements. Panasonic’s PAN1780includes an SoC. One feature of the device is the secure update process for updating the firmware (“system software”) of the SoC (“embedded system”).

53. Panasonic’s PAN1780 includes two sections of firmware: an Immutable section and an Updateable section. The Updateable section of the firmware receives updates over a network and is automatically updated without any physical intervention.



1 PSA overview

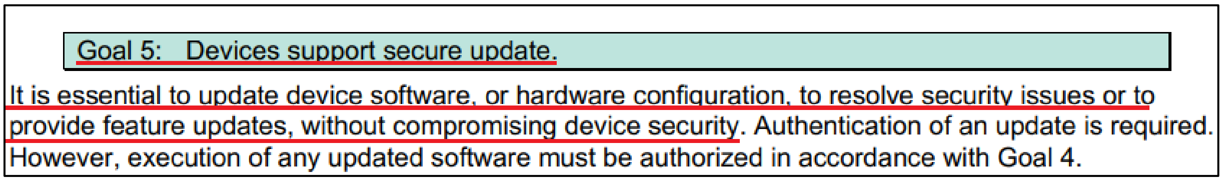
The Platform Security Architecture (PSA) is a framework for securing devices. Though the focus is on local network or internet connected devices, many aspects are relevant for non-connected devices. PSA includes a holistic set of deliverables, including Threat Models and Security Analyses documentation, hardware and firmware architecture specifications, APIs, and an API test suite. Together with an open source reference implementation, PSA enables consistent design-in at the right level of security. PSA-compliant products may go through a security evaluation and become PSA Certified™.

PSA builds upon best practice from across the industry and is aimed at different entities throughout the supply chain, from chip designers and device developers to cloud and network infrastructure providers and software vendors.

This document defines the overall security elements for designing and deploying trusted PSA-compliant devices within ecosystems. This document is the top-level document for the other PSA specifications and defines common language, high-level robustness rules, and models.

PSA is anchored in a set of core security goals, which provide a high-level, abstract, way to think about the essential features that are necessary to secure and establish trust. Abstraction allows these goals to be applied as required, for example, to an end user connected device, a hardware component, a software component, or a service. In describing the goals, the term *device* is used to represent any entity that must be secure and trustworthy.

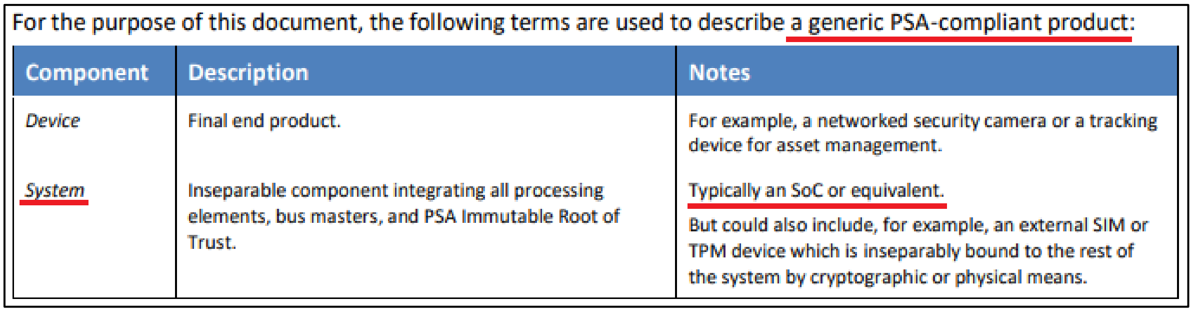
Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 1).



Goal 5: Devices support secure update.

It is essential to update device software, or hardware configuration, to resolve security issues or to provide feature updates, without compromising device security. Authentication of an update is required. However, execution of any updated software must be authorized in accordance with Goal 4.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 2).

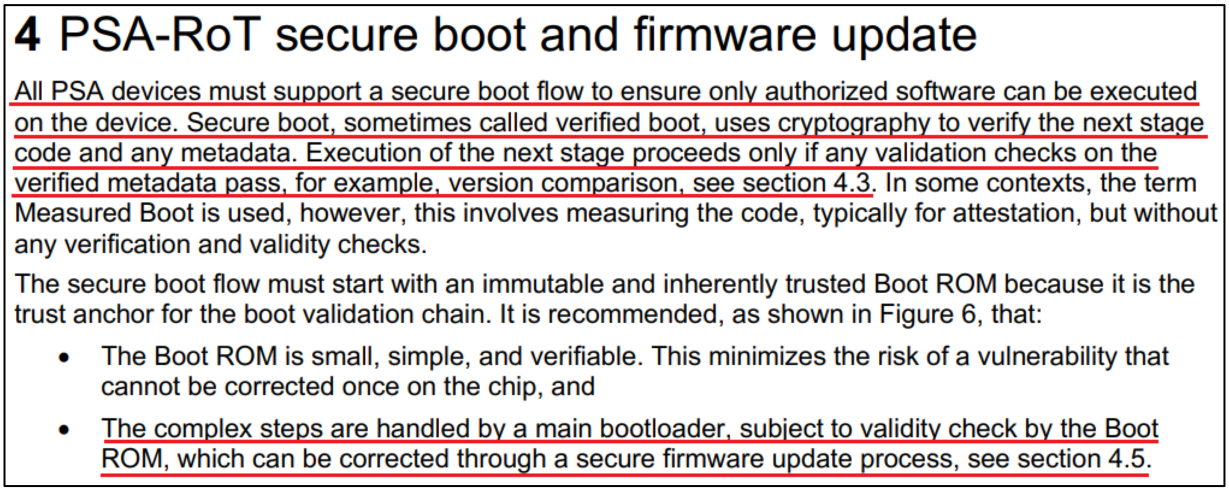


For the purpose of this document, the following terms are used to describe a generic PSA-compliant product:

Component	Description	Notes
Device	Final end product.	For example, a networked security camera or a tracking device for asset management.
System	Inseparable component integrating all processing elements, bus masters, and PSA Immutable Root of Trust.	Typically an SoC or equivalent. But could also include, for example, an external SIM or TPM device which is inseparably bound to the rest of the system by cryptographic or physical means.

Source:

https://armkeil.blob.core.windows.net/developer/Files/pdf/PlatformSecurityArchitecture/Architect/DEN0079-PSA_SM_ALPHA-02.pdf (Page 18).



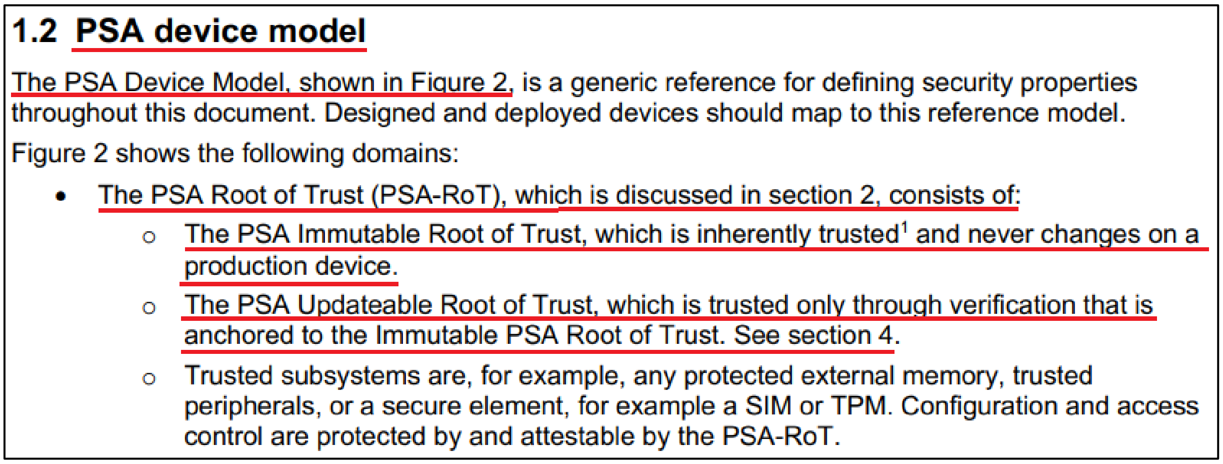
4 PSA-RoT secure boot and firmware update

All PSA devices must support a secure boot flow to ensure only authorized software can be executed on the device. Secure boot, sometimes called verified boot, uses cryptography to verify the next stage code and any metadata. Execution of the next stage proceeds only if any validation checks on the verified metadata pass, for example, version comparison, see section 4.3. In some contexts, the term Measured Boot is used, however, this involves measuring the code, typically for attestation, but without any verification and validity checks.

The secure boot flow must start with an immutable and inherently trusted Boot ROM because it is the trust anchor for the boot validation chain. It is recommended, as shown in Figure 6, that:

- The Boot ROM is small, simple, and verifiable. This minimizes the risk of a vulnerability that cannot be corrected once on the chip, and
- The complex steps are handled by a main bootloader, subject to validity check by the Boot ROM, which can be corrected through a secure firmware update process, see section 4.5.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 14).



1.2 PSA device model

The PSA Device Model, shown in Figure 2, is a generic reference for defining security properties throughout this document. Designed and deployed devices should map to this reference model.

Figure 2 shows the following domains:

- The PSA Root of Trust (PSA-RoT), which is discussed in section 2, consists of:
  - The PSA Immutable Root of Trust, which is inherently trusted[1] and never changes on a production device.
  - The PSA Updateable Root of Trust, which is trusted only through verification that is anchored to the Immutable PSA Root of Trust. See section 4.
  - Trusted subsystems are, for example, any protected external memory, trusted peripherals, or a secure element, for example a SIM or TPM. Configuration and access control are protected by and attestable by the PSA-RoT.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 4).

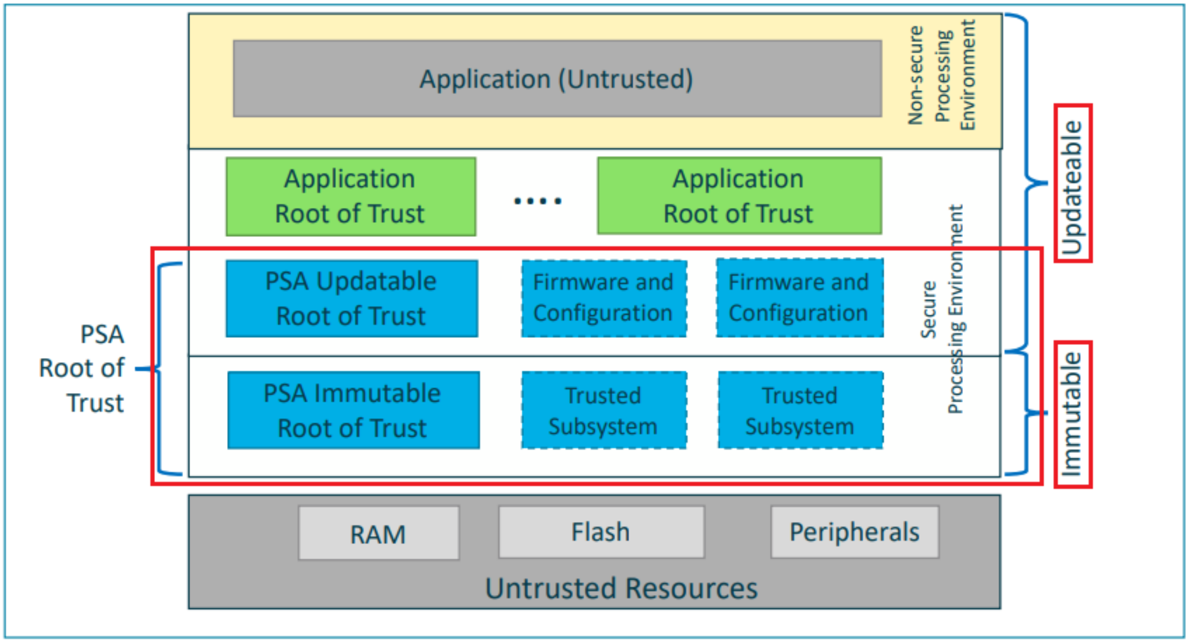
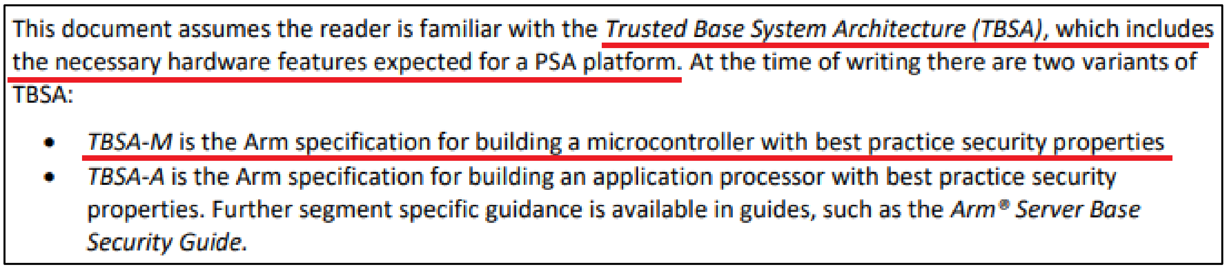

Application (Untrusted)
Non-secure Processing Environment
Application Root of Trust
....
Application Root of Trust
PSA Updatable Root of Trust
Firmware and Configuration
Firmware and Configuration
Secure Processing Environment
PSA Root of Trust
PSA Immutable Root of Trust
Trusted Subsystem
Trusted Subsystem
Updateable
Immutable
RAM
Flash
Peripherals
Untrusted Resources

Figure 2: PSA device model

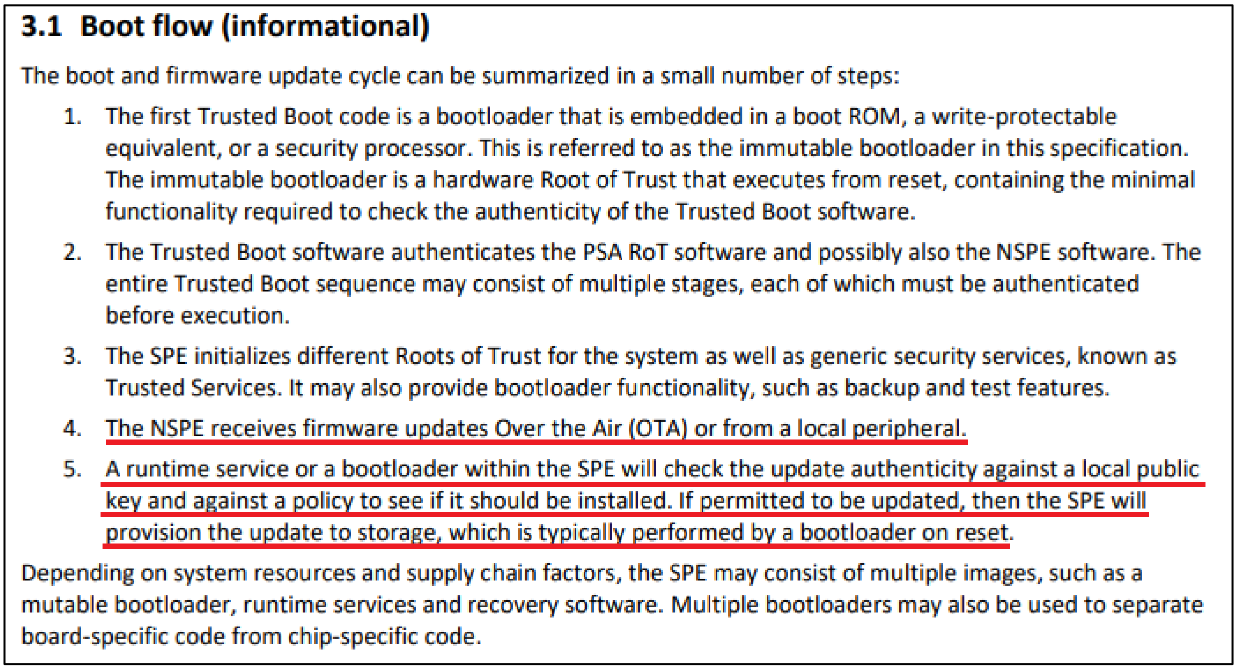
Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 5).



This document assumes the reader is familiar with the *Trusted Base System Architecture (TBSA)*, which includes the necessary hardware features expected for a PSA platform. At the time of writing there are two variants of TBSA:

- *TBSA-M* is the Arm specification for building a microcontroller with best practice security properties
- *TBSA-A* is the Arm specification for building an application processor with best practice security properties. Further segment specific guidance is available in guides, such as the *Arm® Server Base Security Guide*.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Pages 13 and 14).



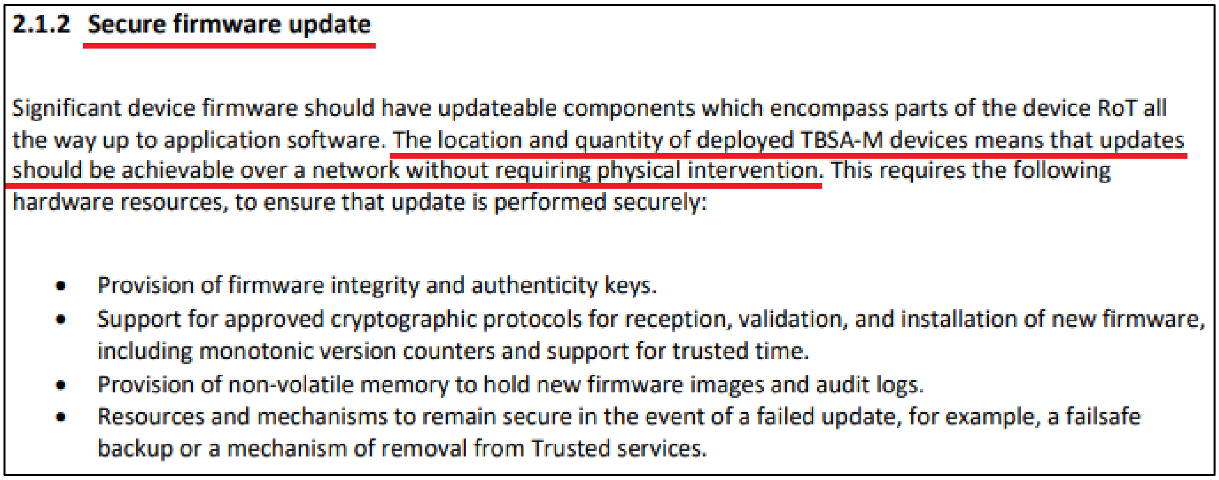
3.1 Boot flow (informational)

The boot and firmware update cycle can be summarized in a small number of steps:

1. The first Trusted Boot code is a bootloader that is embedded in a boot ROM, a write-protectable equivalent, or a security processor. This is referred to as the immutable bootloader in this specification. The immutable bootloader is a hardware Root of Trust that executes from reset, containing the minimal functionality required to check the authenticity of the Trusted Boot software.
2. The Trusted Boot software authenticates the PSA RoT software and possibly also the NSPE software. The entire Trusted Boot sequence may consist of multiple stages, each of which must be authenticated before execution.
3. The SPE initializes different Roots of Trust for the system as well as generic security services, known as Trusted Services. It may also provide bootloader functionality, such as backup and test features.
4. The NSPE receives firmware updates Over the Air (OTA) or from a local peripheral.
5. A runtime service or a bootloader within the SPE will check the update authenticity against a local public key and against a policy to see if it should be installed. If permitted to be updated, then the SPE will provision the update to storage, which is typically performed by a bootloader on reset.

Depending on system resources and supply chain factors, the SPE may consist of multiple images, such as a mutable bootloader, runtime services and recovery software. Multiple bootloaders may also be used to separate board-specific code from chip-specific code.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Page 18).



2.1.2 Secure firmware update

Significant device firmware should have updateable components which encompass parts of the device RoT all the way up to application software. The location and quantity of deployed TBSA-M devices means that updates should be achievable over a network without requiring physical intervention. This requires the following hardware resources, to ensure that update is performed securely:

- Provision of firmware integrity and authenticity keys.
- Support for approved cryptographic protocols for reception, validation, and installation of new firmware, including monotonic version counters and support for trusted time.
- Provision of non-volatile memory to hold new firmware images and audit logs.
- Resources and mechanisms to remain secure in the event of a failed update, for example, a failsafe backup or a mechanism of removal from Trusted services.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 18).

54. Panasonic's PAN1780 includes a first storage device for storing a first system software and a boot image.

55. For example, Panasonic's PAN1780 includes two sections of firmware: an Immutable section and an Updateable section. The Immutable section of the firmware contains code that is executed immediately after powering on the system. Accordingly, the Immutable section of the firmware is the "boot image" of device. Further, the Immutable section ("boot image") and Updateable section ("first system software") of firmware are stored in the embedded flash ("first storage device").

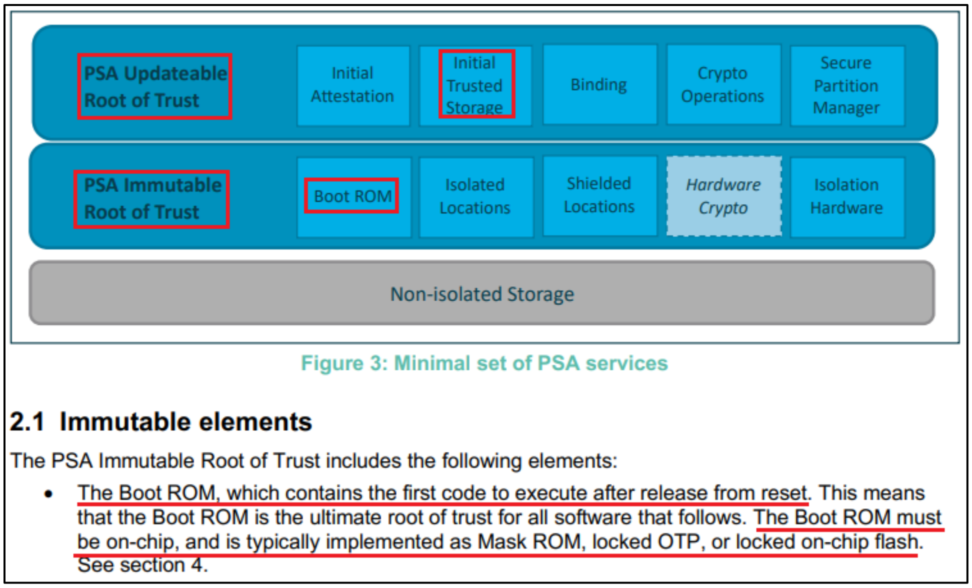


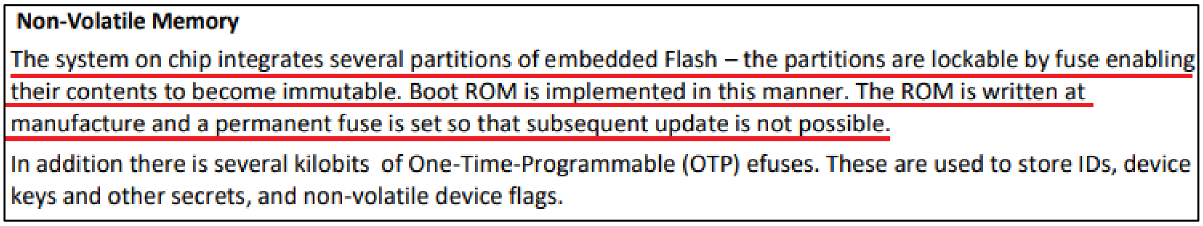
Figure 3: Minimal set of PSA services

2.1 Immutable elements

The PSA Immutable Root of Trust includes the following elements:

- The Boot ROM, which contains the first code to execute after release from reset. This means that the Boot ROM is the ultimate root of trust for all software that follows. The Boot ROM must be on-chip, and is typically implemented as Mask ROM, locked OTP, or locked on-chip flash. See section 4.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 6).



Non-Volatile Memory

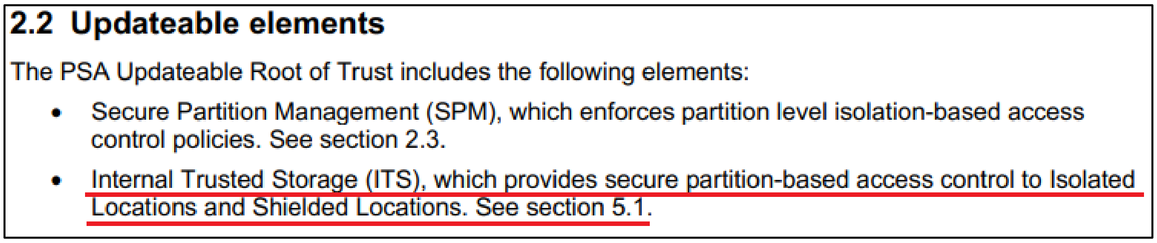
The system on chip integrates several partitions of embedded Flash – the partitions are lockable by fuse enabling their contents to become immutable. Boot ROM is implemented in this manner. The ROM is written at manufacture and a permanent fuse is set so that subsequent update is not possible.

In addition there is several kilobits of One-Time-Programmable (OTP) efuses. These are used to store IDs, device keys and other secrets, and non-volatile device flags.

Source:

https://armkeil.blob.core.windows.net/developer/Files/pdf/PlatformSecurityArchitecture/Architect/DEN0079-PSA_SM_ALPHA-02.pdf (Page 70).

Citation 12:

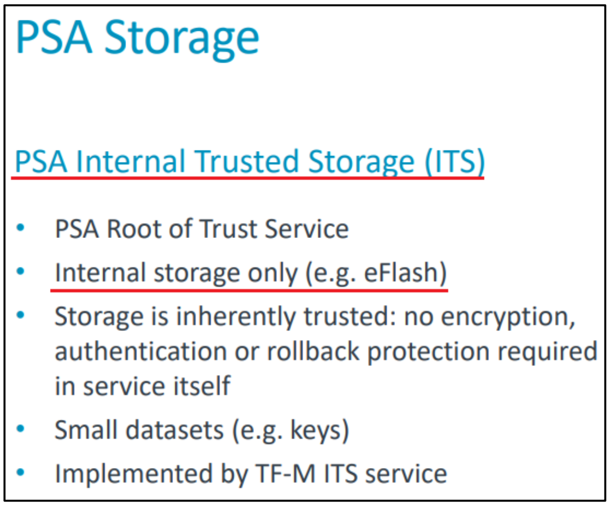


2.2 Updateable elements

The PSA Updateable Root of Trust includes the following elements:

- Secure Partition Management (SPM), which enforces partition level isolation-based access control policies. See section 2.3.
- Internal Trusted Storage (ITS), which provides secure partition-based access control to Isolated Locations and Shielded Locations. See section 5.1.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 7).

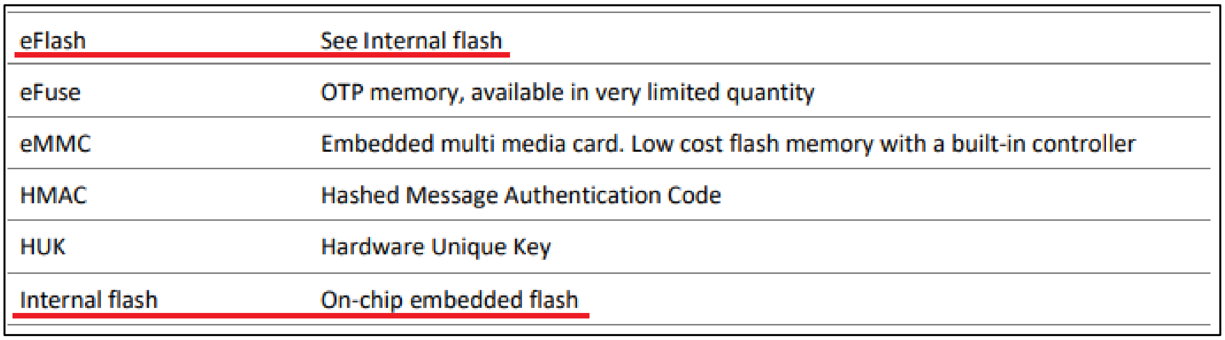


PSA Storage

PSA Internal Trusted Storage (ITS)

- PSA Root of Trust Service
- Internal storage only (e.g. eFlash)
- Storage is inherently trusted: no encryption, authentication or rollback protection required in service itself
- Small datasets (e.g. keys)
- Implemented by TF-M ITS service

Source: https://www.trustedfirmware.org/docs/tfm_secure_storage_tech_forum.pdf (Page 3).



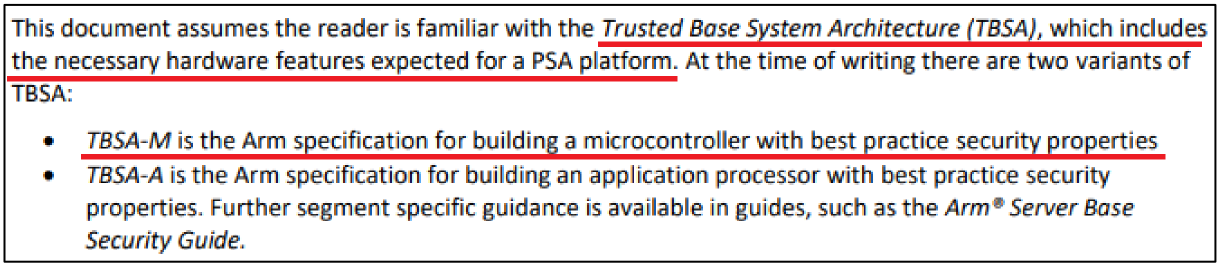
eFlash	See Internal flash
eFuse	OTP memory, available in very limited quantity
eMMC	Embedded multi media card. Low cost flash memory with a built-in controller
HMAC	Hashed Message Authentication Code
HUK	Hardware Unique Key
Internal flash	On-chip embedded flash

Source: https://pages.arm.com/rs/312-SAX-488/images/DEN0072-PSA_TBFU_1.0-bet1.pdf (Page 9).

56. Panasonic's PAN1780 includes a micro-controller that is coupled to the first storage device for respectively transforming the first system software and the boot image into a system code and a boot code. The micro-controller orderly executes the boot code and the system code to control booting of the embedded system.

57. For example, the device supports Trusted Base System Architectures like TBSA-M and TBSA-A. TBSA-M specifies the architecture for products including ARM micro-controllers. The embedded flash of Panasonic's PAN1780's micro-controller includes two sections of firmware images: an Immutable boot image ("boot image") and an Updateable firmware image ("first system software").

58. Before booting the system, a small program called boot loader (i.e., start-up code) uses linker files to map sections of the Immutable and Updateable firmware images to different parts of the memory for execution. That is, the firmware images are stored in one memory location before loading and in another memory location during execution. This is accomplished through ARM's linker files, which provide information about the mapping of different sections of the firmware images to different sections of memory. Once the mapping into memory is complete, the micro-controller's processor is then capable of executing the mapped Immutable firmware ("boot code") and the mapped Updateable firmware ("system code"). Accordingly, the micro-controller's processor transforms the "boot image" and the "system software" into memory-mapped executables (i.e., boot code and system code, respectively).



This document assumes the reader is familiar with the *Trusted Base System Architecture (TBSA)*, which includes the necessary hardware features expected for a PSA platform. At the time of writing there are two variants of TBSA:

- *TBSA-M* is the Arm specification for building a microcontroller with best practice security properties
- *TBSA-A* is the Arm specification for building an application processor with best practice security properties. Further segment specific guidance is available in guides, such as the *Arm® Server Base Security Guide*.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Pages 13 and 14).

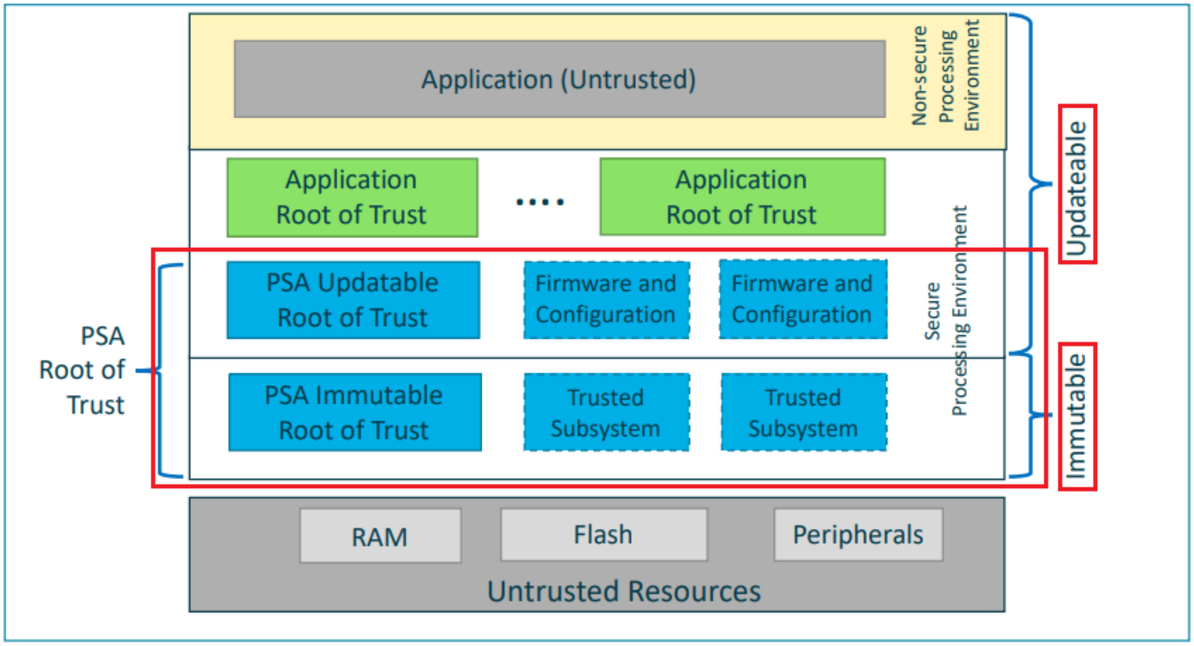
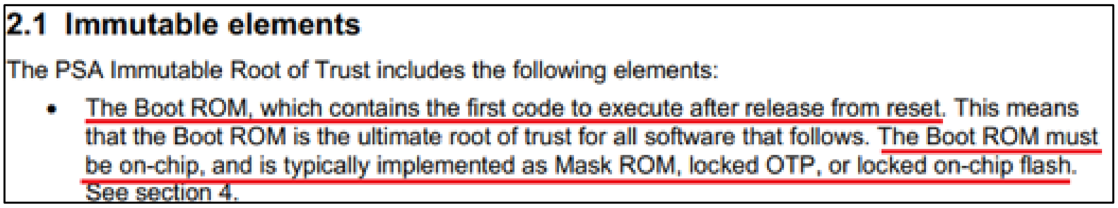

Application (Untrusted)
Non-secure Processing Environment
Application Root of Trust
....
Application Root of Trust
PSA Updatable Root of Trust
Firmware and Configuration
Firmware and Configuration
Secure Processing Environment
PSA Root of Trust
PSA Immutable Root of Trust
Trusted Subsystem
Trusted Subsystem
Updateable
Immutable
RAM
Flash
Peripherals
Untrusted Resources

Figure 2: PSA device model

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 5).

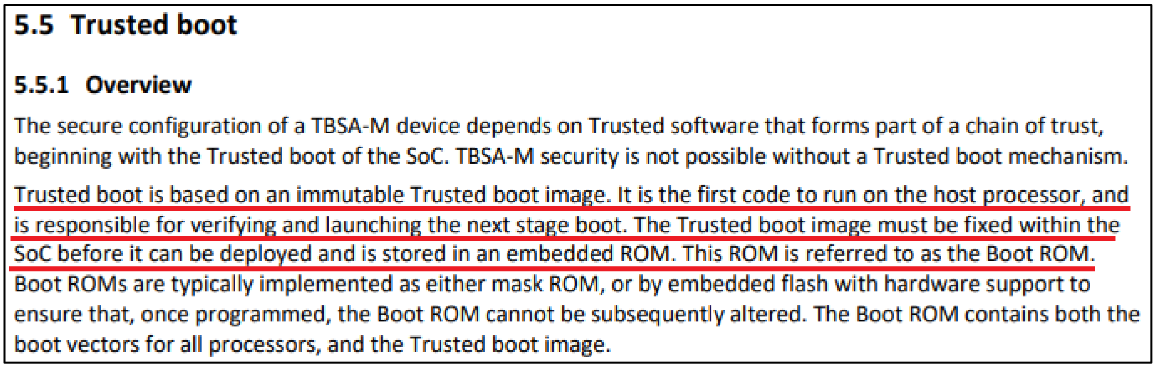


2.1 Immutable elements

The PSA Immutable Root of Trust includes the following elements:

- The Boot ROM, which contains the first code to execute after release from reset. This means that the Boot ROM is the ultimate root of trust for all software that follows. The Boot ROM must be on-chip, and is typically implemented as Mask ROM, locked OTP, or locked on-chip flash. See section 4.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 6).



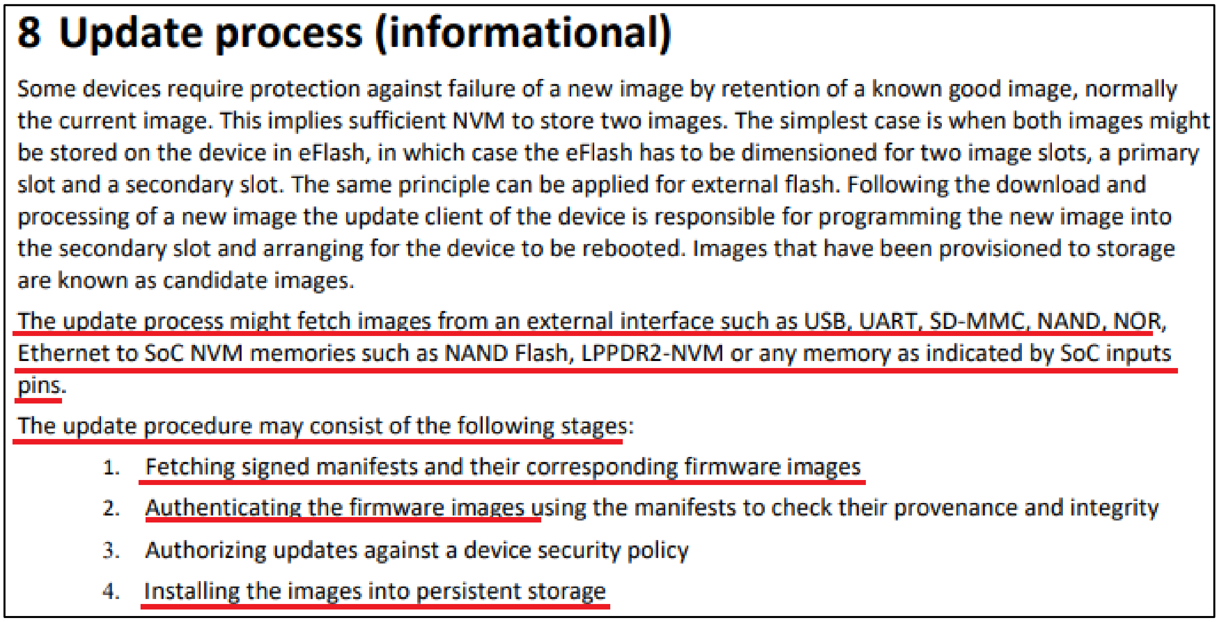
5.5 Trusted boot

5.5.1 Overview

The secure configuration of a TBSA-M device depends on Trusted software that forms part of a chain of trust, beginning with the Trusted boot of the SoC. TBSA-M security is not possible without a Trusted boot mechanism.

Trusted boot is based on an immutable Trusted boot image. It is the first code to run on the host processor, and is responsible for verifying and launching the next stage boot. The Trusted boot image must be fixed within the SoC before it can be deployed and is stored in an embedded ROM. This ROM is referred to as the Boot ROM. Boot ROMs are typically implemented as either mask ROM, or by embedded flash with hardware support to ensure that, once programmed, the Boot ROM cannot be subsequently altered. The Boot ROM contains both the boot vectors for all processors, and the Trusted boot image.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 50).



8 Update process (informational)

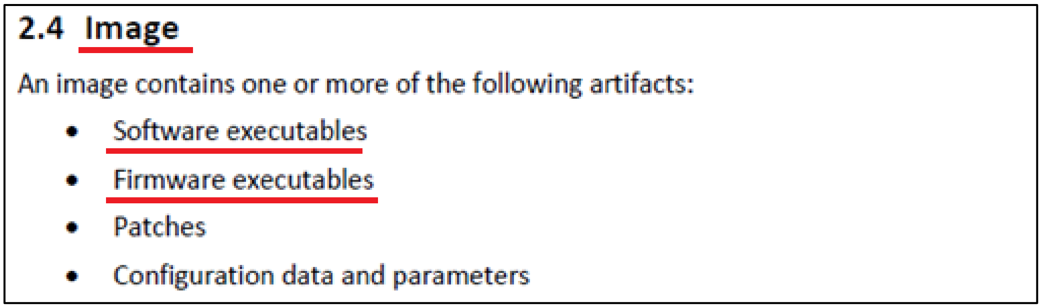
Some devices require protection against failure of a new image by retention of a known good image, normally the current image. This implies sufficient NVM to store two images. The simplest case is when both images might be stored on the device in eFlash, in which case the eFlash has to be dimensioned for two image slots, a primary slot and a secondary slot. The same principle can be applied for external flash. Following the download and processing of a new image the update client of the device is responsible for programming the new image into the secondary slot and arranging for the device to be rebooted. Images that have been provisioned to storage are known as candidate images.

The update process might fetch images from an external interface such as USB, UART, SD-MMC, NAND, NOR, Ethernet to SoC NVM memories such as NAND Flash, LPPDR2-NVM or any memory as indicated by SoC inputs pins.

The update procedure may consist of the following stages:

1. Fetching signed manifests and their corresponding firmware images
2. Authenticating the firmware images using the manifests to check their provenance and integrity
3. Authorizing updates against a device security policy
4. Installing the images into persistent storage

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Page 42).

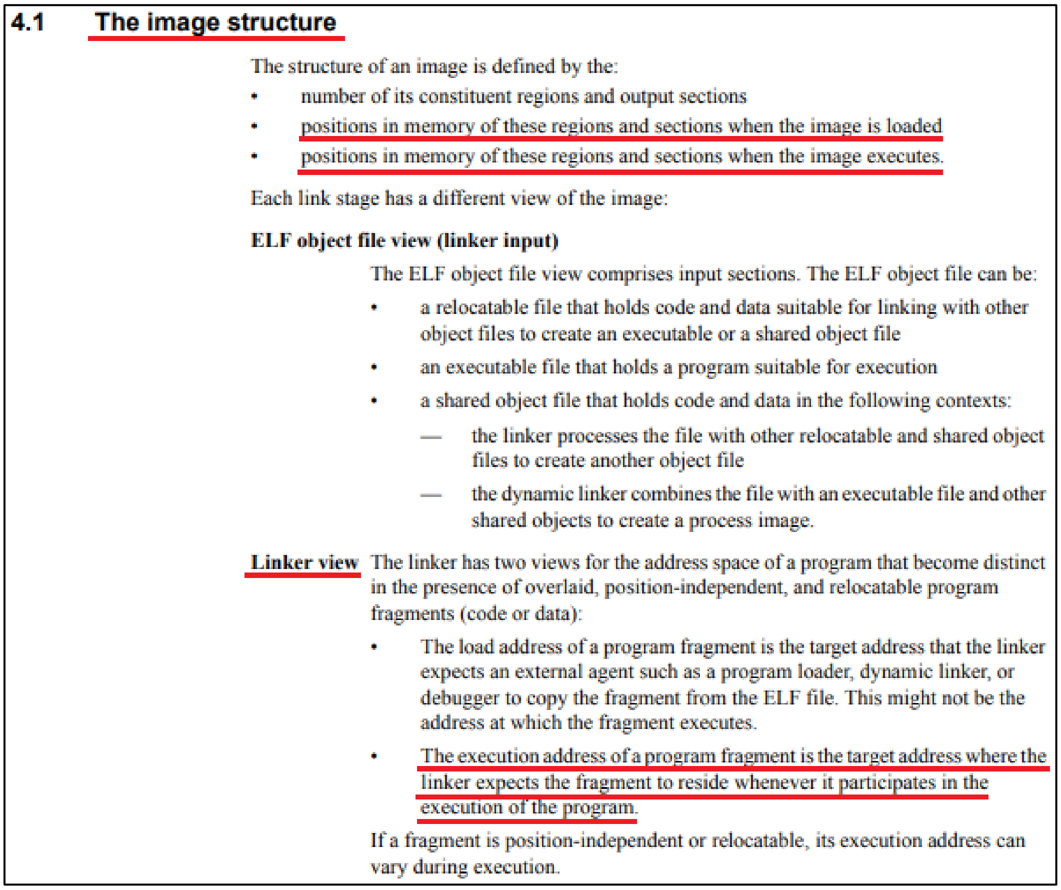


2.4 Image

An image contains one or more of the following artifacts:

- Software executables
- Firmware executables
- Patches
- Configuration data and parameters

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Page 17).



4.1 The image structure

The structure of an image is defined by the:

- number of its constituent regions and output sections
- positions in memory of these regions and sections when the image is loaded
- positions in memory of these regions and sections when the image executes.

Each link stage has a different view of the image:

ELF object file view (linker input)

The ELF object file view comprises input sections. The ELF object file can be:

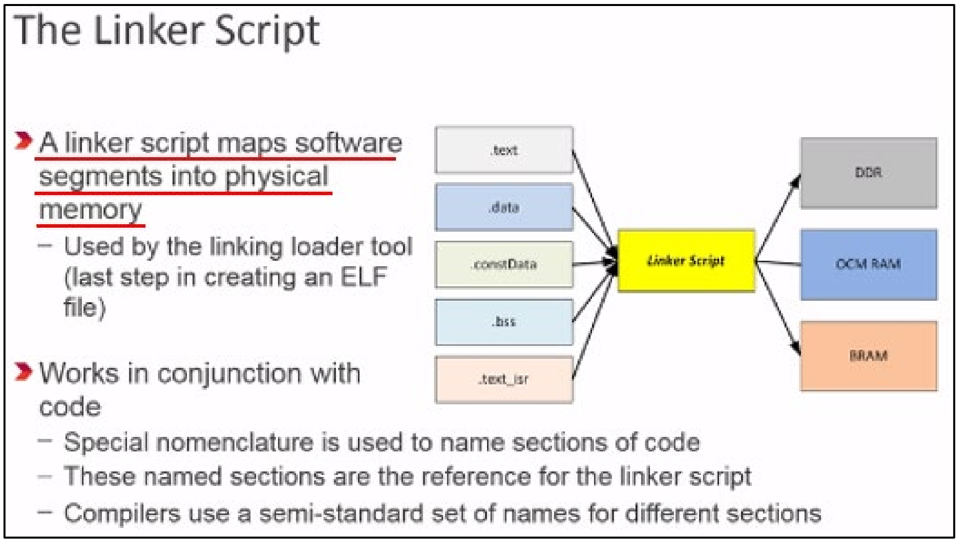
- a relocatable file that holds code and data suitable for linking with other object files to create an executable or a shared object file
- an executable file that holds a program suitable for execution
- a shared object file that holds code and data in the following contexts:
  - — the linker processes the file with other relocatable and shared object files to create another object file
  - — the dynamic linker combines the file with an executable file and other shared objects to create a process image.

Linker view The linker has two views for the address space of a program that become distinct in the presence of overlaid, position-independent, and relocatable program fragments (code or data):

- The load address of a program fragment is the target address that the linker expects an external agent such as a program loader, dynamic linker, or debugger to copy the fragment from the ELF file. This might not be the address at which the fragment executes.
- The execution address of a program fragment is the target address where the linker expects the fragment to reside whenever it participates in the execution of the program.

If a fragment is position-independent or relocatable, its execution address can vary during execution.

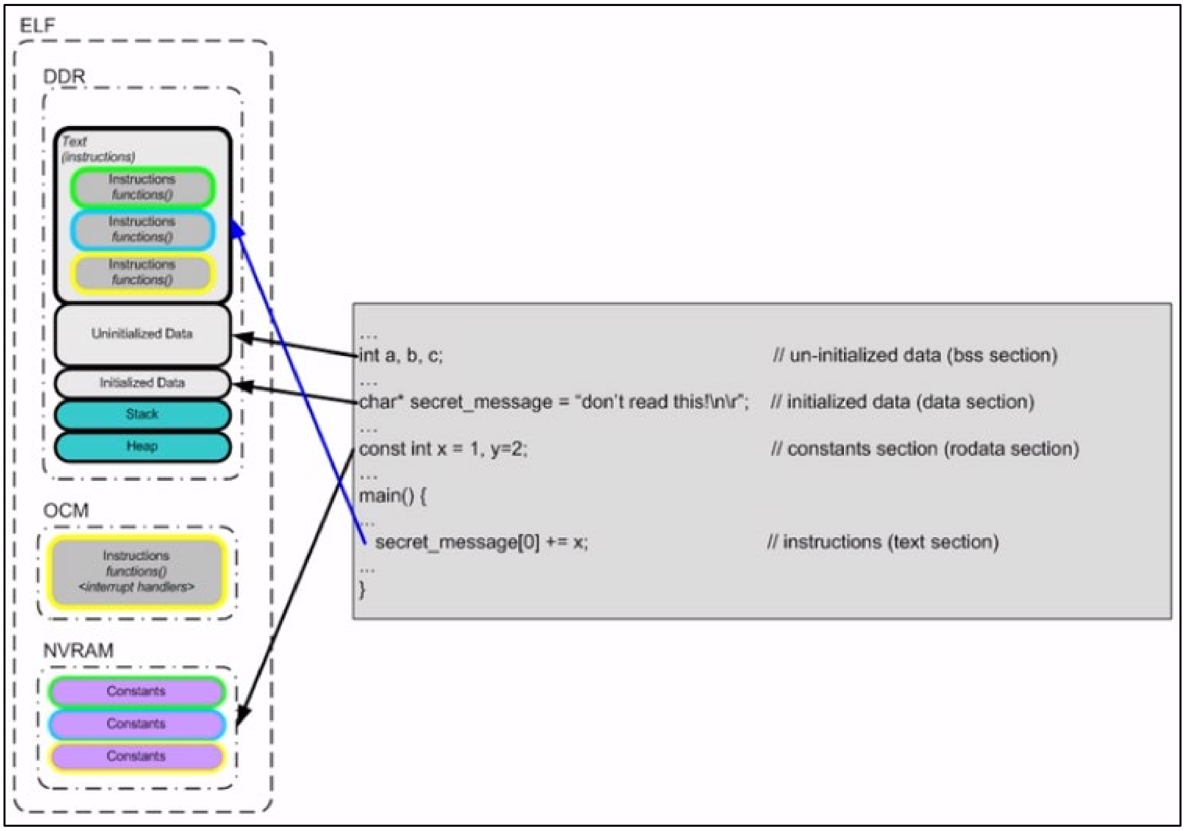
Source: https://documentation-service.arm.com/static/5ea19a939931941038de91a1?token= (Page 32).



The Linker Script

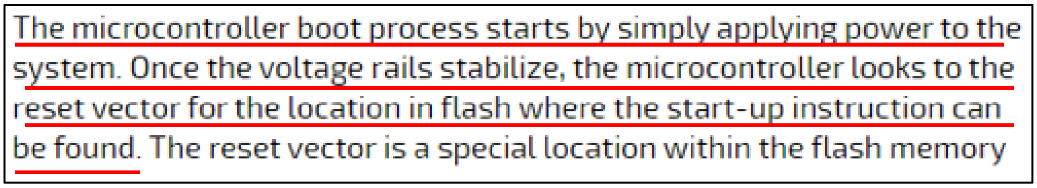
- A linker script maps software segments into physical memory
  - Used by the linking loader tool (last step in creating an ELF file)
- Works in conjunction with code
  - Special nomenclature is used to name sections of code
  - These named sections are the reference for the linker script
  - Compilers use a semi-standard set of names for different sections

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



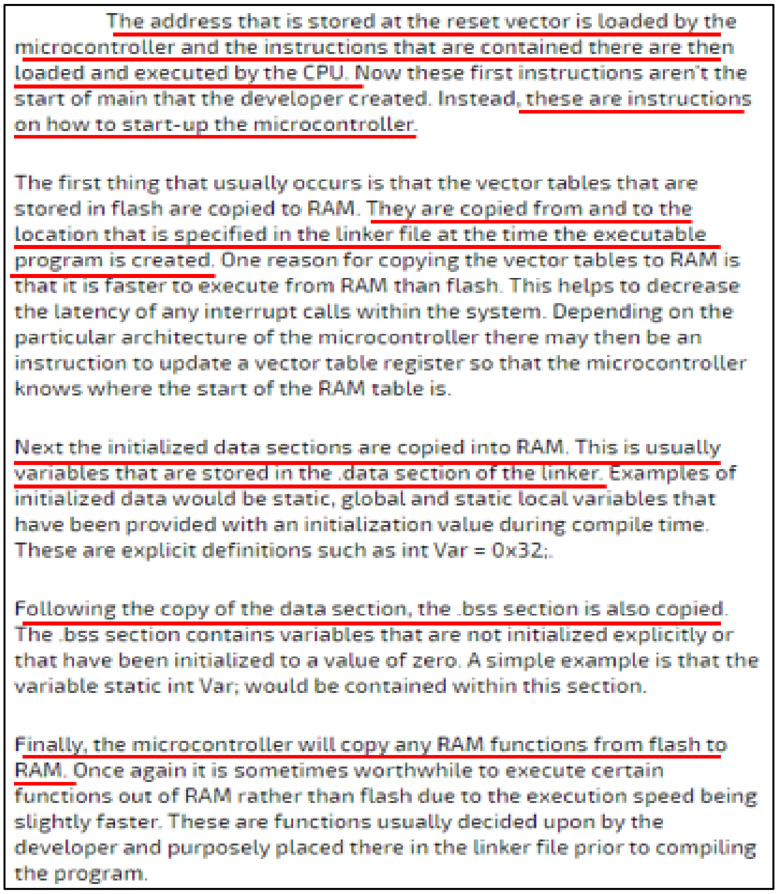
```
...
int a, b, c;                                   // un-initialized data (bss section)
...
char* secret_message = "don't read this!\n\r";  // initialized data (data section)
...
const int x = 1, y=2;                          // constants section (rodata section)
...
main() {
...
  secret_message[0] += x;                      // instructions (text section)
...
}
```

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



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.

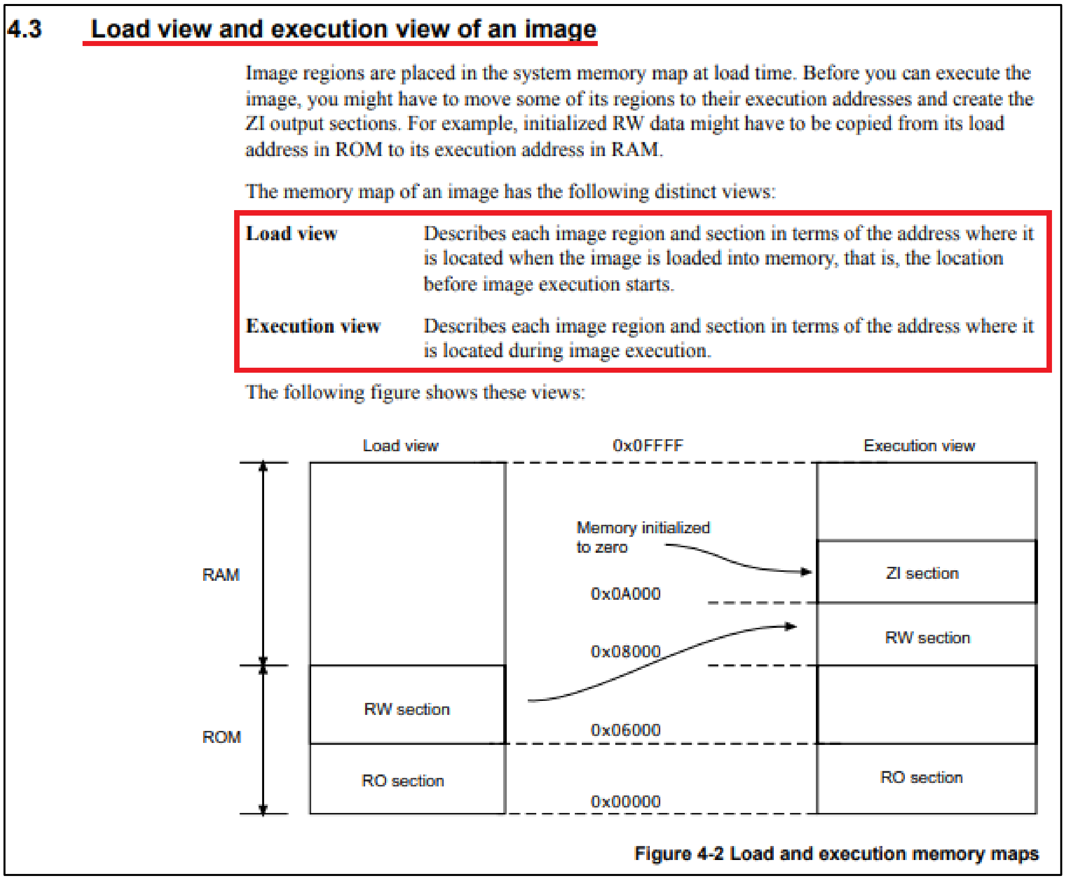
The first thing that usually occurs is that the vector tables that are stored in flash are copied to RAM. They are copied from and to the location that is specified in the linker file at the time the executable program is created. One reason for copying the vector tables to RAM is that it is faster to 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/



4.3 Load view and execution view of an image

Image regions are placed in the system memory map at load time. Before you can execute the image, you might have to move some of its regions to their execution addresses and create the ZI output sections. For example, initialized RW data might have to be copied from its load address in ROM to its execution address in RAM.

The memory map of an image has the following distinct views:

Load view Describes each image region and section in terms of the address where it is located when the image is loaded into memory, that is, the location before image execution starts.

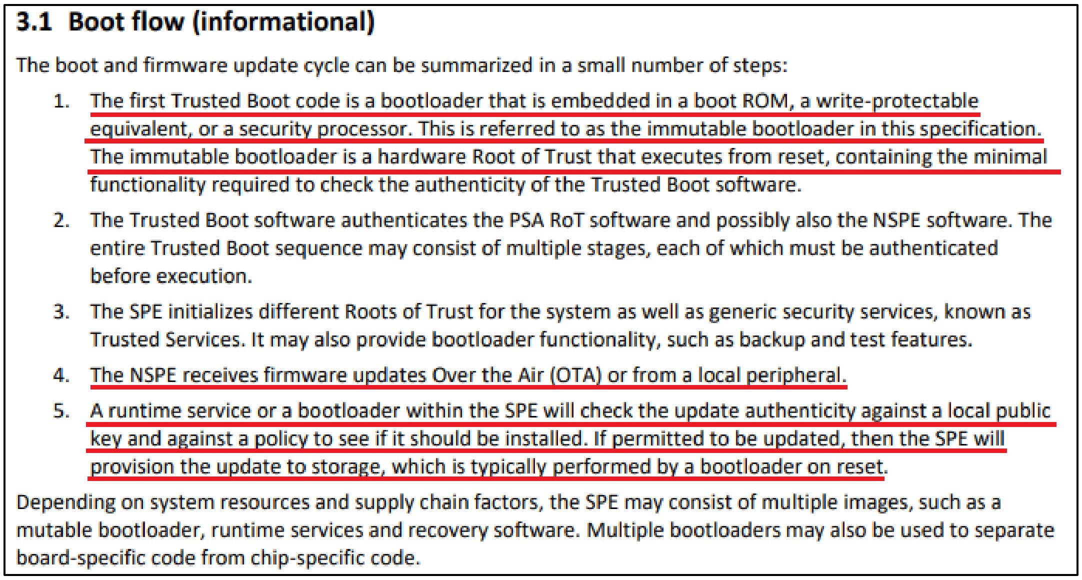
Execution view Describes each image region and section in terms of the address where it is located during image execution.

The following figure shows these views:

Figure 4-2 Load and execution memory maps

Source: https://documentation-service.arm.com/static/5ea19a939931941038de91a1?token= (Page 32).

59. During initialization and booting of Panasonic’s PAN1780, the mapped Immutable firmware section (“boot code”) is executed first. The Immutable firmware section then runs the mapped Updateable firmware (“system code”). Accordingly, the boot code and system code are orderly executed by the microcontroller to control the booting of the embedded system.



3.1 Boot flow (informational)

The boot and firmware update cycle can be summarized in a small number of steps:

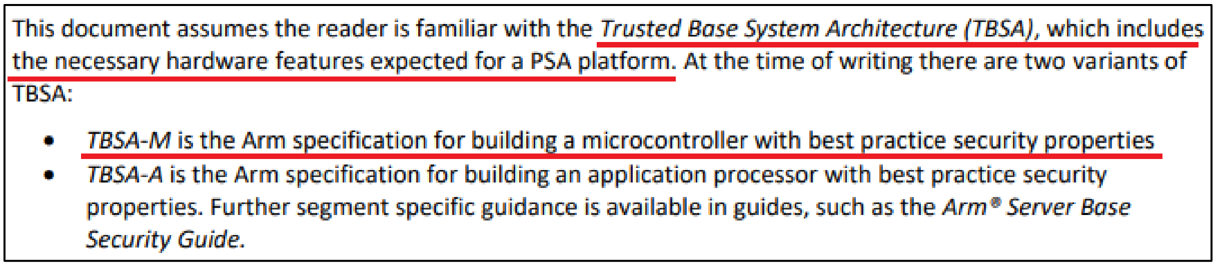
1. The first Trusted Boot code is a bootloader that is embedded in a boot ROM, a write-protectable equivalent, or a security processor. This is referred to as the immutable bootloader in this specification. The immutable bootloader is a hardware Root of Trust that executes from reset, containing the minimal functionality required to check the authenticity of the Trusted Boot software.
2. The Trusted Boot software authenticates the PSA RoT software and possibly also the NSPE software. The entire Trusted Boot sequence may consist of multiple stages, each of which must be authenticated before execution.
3. The SPE initializes different Roots of Trust for the system as well as generic security services, known as Trusted Services. It may also provide bootloader functionality, such as backup and test features.
4. The NSPE receives firmware updates Over the Air (OTA) or from a local peripheral.
5. A runtime service or a bootloader within the SPE will check the update authenticity against a local public key and against a policy to see if it should be installed. If permitted to be updated, then the SPE will provision the update to storage, which is typically performed by a bootloader on reset.

Depending on system resources and supply chain factors, the SPE may consist of multiple images, such as a mutable bootloader, runtime services and recovery software. Multiple bootloaders may also be used to separate board-specific code from chip-specific code.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Page 18).

60. Panasonic's PAN1780 includes a connecting interface that is coupled to the micro-controller and to an external data storage device through a data transmission media. The external data storage device stores a second system software.

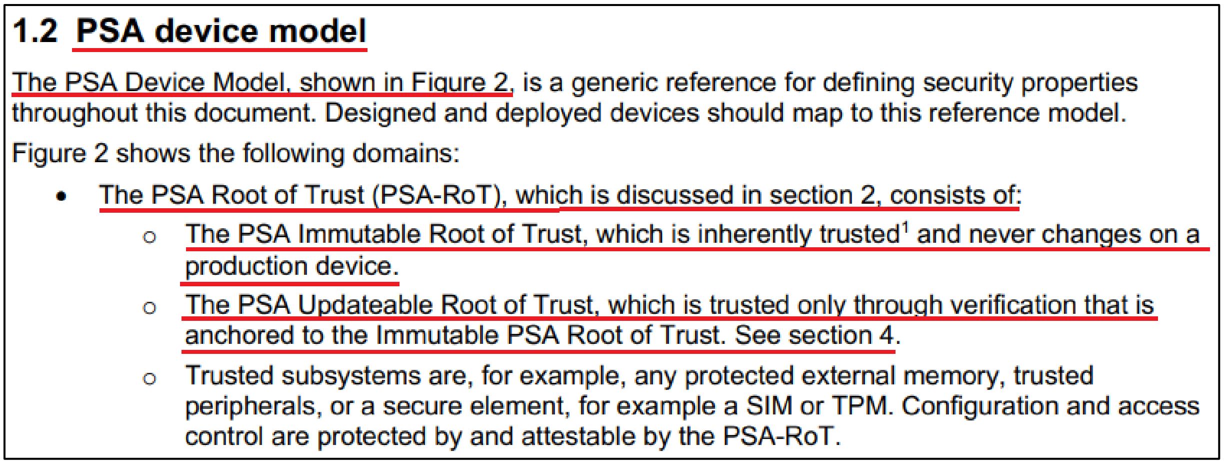
61. For example, the device supports Trusted Base System Architectures like TBSA-M and TBSA-A. TBSA-M specifies the architecture for products that include ARM micro-controllers. Panasonic's PAN1780 includes two sections of firmware: an Immutable section and an Updateable section. The firmware update of the Updateable section ("second system software") is sent over-the-air via a remote server or by an external local peripheral device ("external data storage device"). The update can be received over-the-air or through an external interface such as USB, UART, SD-MMC, Ethernet, etc. ("connecting interface"). The external interface is capable of being connected to the external data storage device and the micro-controller of the PAN1780 to facilitate the firmware update process.



This document assumes the reader is familiar with the *Trusted Base System Architecture (TBSA)*, which includes the necessary hardware features expected for a PSA platform. At the time of writing there are two variants of TBSA:

- *TBSA-M* is the Arm specification for building a microcontroller with best practice security properties
- *TBSA-A* is the Arm specification for building an application processor with best practice security properties. Further segment specific guidance is available in guides, such as the *Arm® Server Base Security Guide*.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Pages 13 and 14).



1.2 PSA device model

The PSA Device Model, shown in Figure 2, is a generic reference for defining security properties throughout this document. Designed and deployed devices should map to this reference model.

Figure 2 shows the following domains:

- The PSA Root of Trust (PSA-RoT), which is discussed in section 2, consists of:
  - The PSA Immutable Root of Trust, which is inherently trusted[1] and never changes on a production device.
  - The PSA Updateable Root of Trust, which is trusted only through verification that is anchored to the Immutable PSA Root of Trust. See section 4.
  - Trusted subsystems are, for example, any protected external memory, trusted peripherals, or a secure element, for example a SIM or TPM. Configuration and access control are protected by and attestable by the PSA-RoT.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 4).

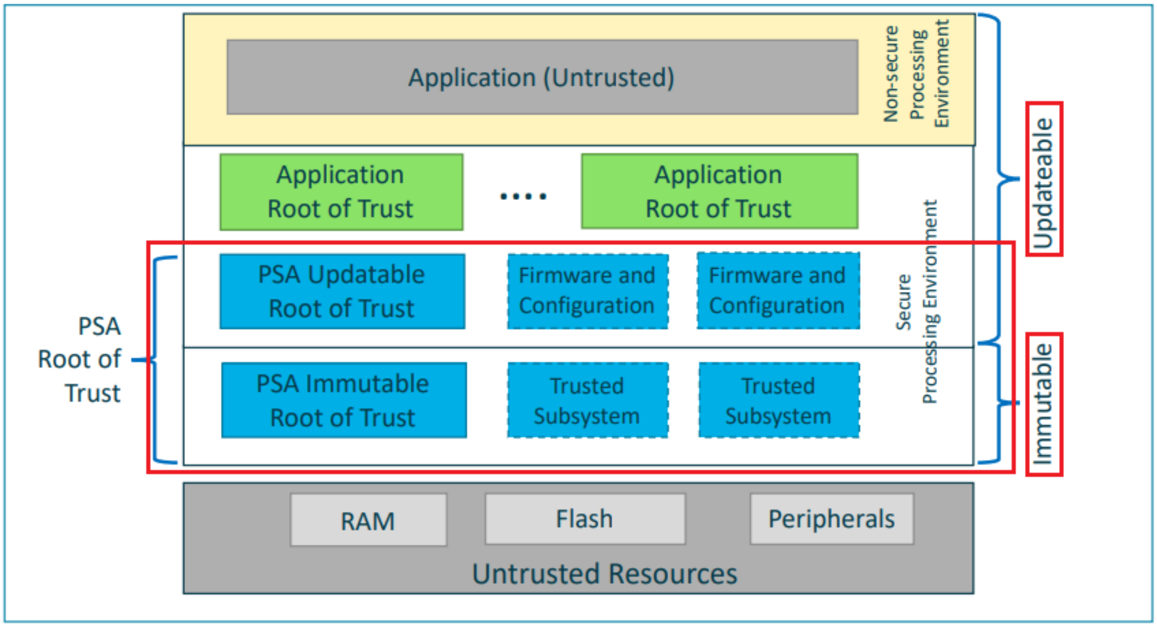
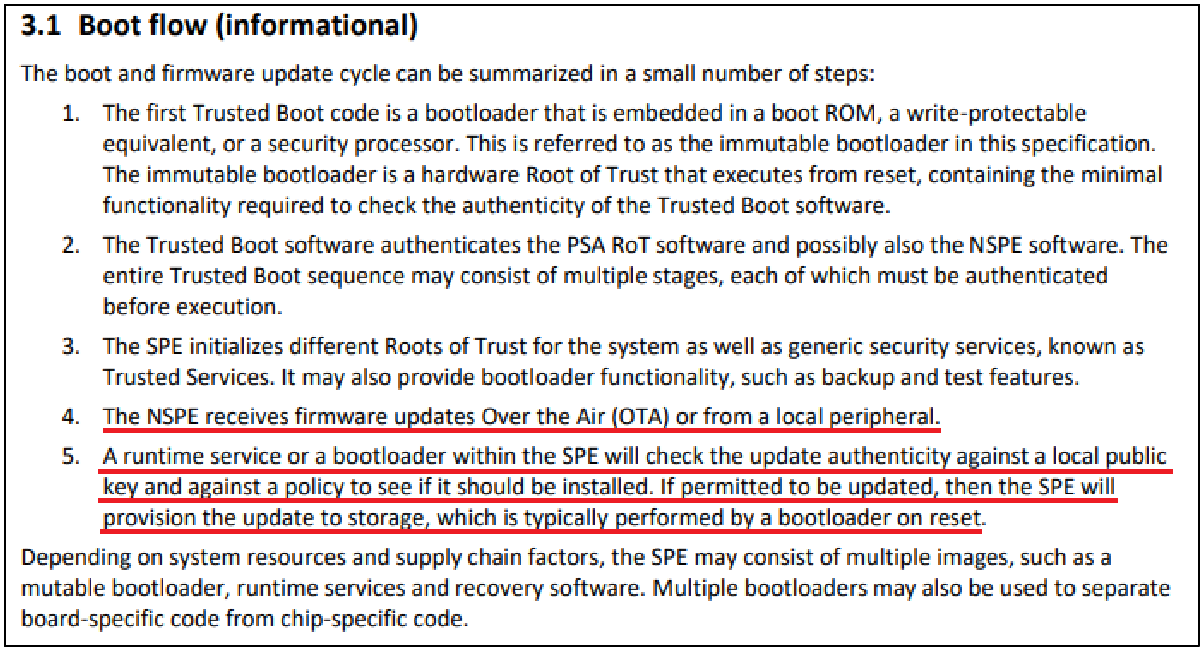

Application (Untrusted)
Non-secure Processing Environment
Application Root of Trust
....
Application Root of Trust
PSA Updatable Root of Trust
Firmware and Configuration
Firmware and Configuration
Secure Processing Environment
PSA Root of Trust
PSA Immutable Root of Trust
Trusted Subsystem
Trusted Subsystem
Updateable
Immutable
RAM
Flash
Peripherals
Untrusted Resources

Figure 2: PSA device model

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 5).



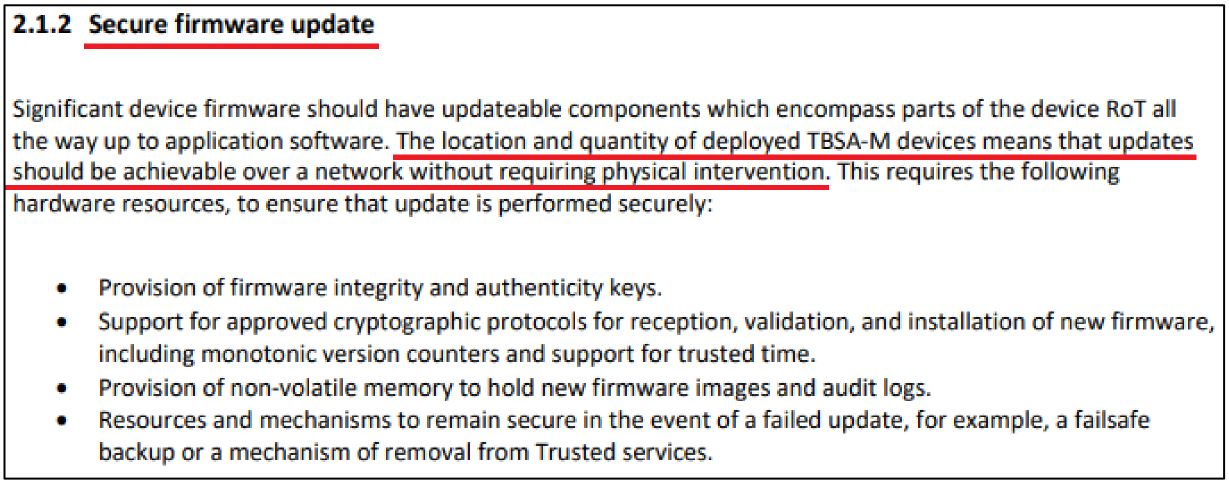
3.1 Boot flow (informational)

The boot and firmware update cycle can be summarized in a small number of steps:

1. The first Trusted Boot code is a bootloader that is embedded in a boot ROM, a write-protectable equivalent, or a security processor. This is referred to as the immutable bootloader in this specification. The immutable bootloader is a hardware Root of Trust that executes from reset, containing the minimal functionality required to check the authenticity of the Trusted Boot software.
2. The Trusted Boot software authenticates the PSA RoT software and possibly also the NSPE software. The entire Trusted Boot sequence may consist of multiple stages, each of which must be authenticated before execution.
3. The SPE initializes different Roots of Trust for the system as well as generic security services, known as Trusted Services. It may also provide bootloader functionality, such as backup and test features.
4. The NSPE receives firmware updates Over the Air (OTA) or from a local peripheral.
5. A runtime service or a bootloader within the SPE will check the update authenticity against a local public key and against a policy to see if it should be installed. If permitted to be updated, then the SPE will provision the update to storage, which is typically performed by a bootloader on reset.

Depending on system resources and supply chain factors, the SPE may consist of multiple images, such as a mutable bootloader, runtime services and recovery software. Multiple bootloaders may also be used to separate board-specific code from chip-specific code.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Page 18).

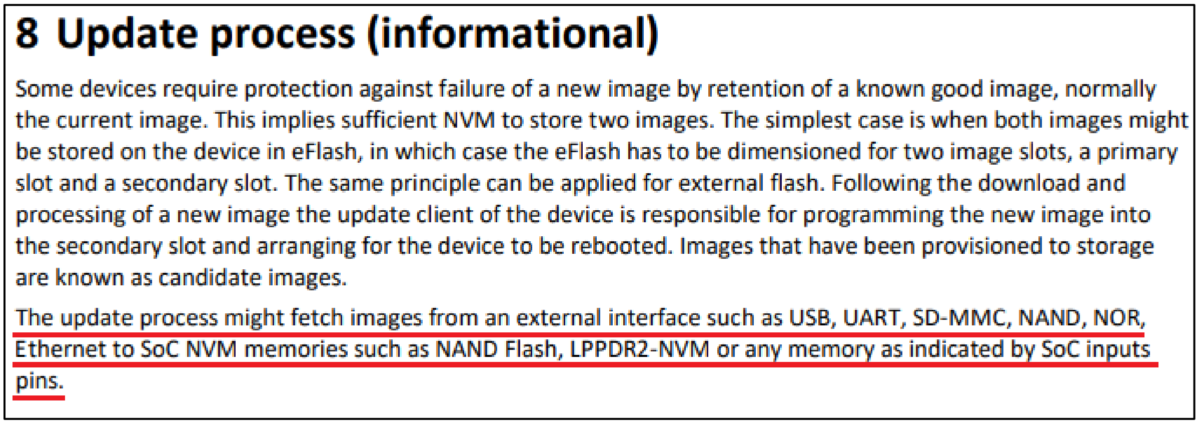


2.1.2 Secure firmware update

Significant device firmware should have updateable components which encompass parts of the device RoT all the way up to application software. The location and quantity of deployed TBSA-M devices means that updates should be achievable over a network without requiring physical intervention. This requires the following hardware resources, to ensure that update is performed securely:

- Provision of firmware integrity and authenticity keys.
- Support for approved cryptographic protocols for reception, validation, and installation of new firmware, including monotonic version counters and support for trusted time.
- Provision of non-volatile memory to hold new firmware images and audit logs.
- Resources and mechanisms to remain secure in the event of a failed update, for example, a failsafe backup or a mechanism of removal from Trusted services.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 18).

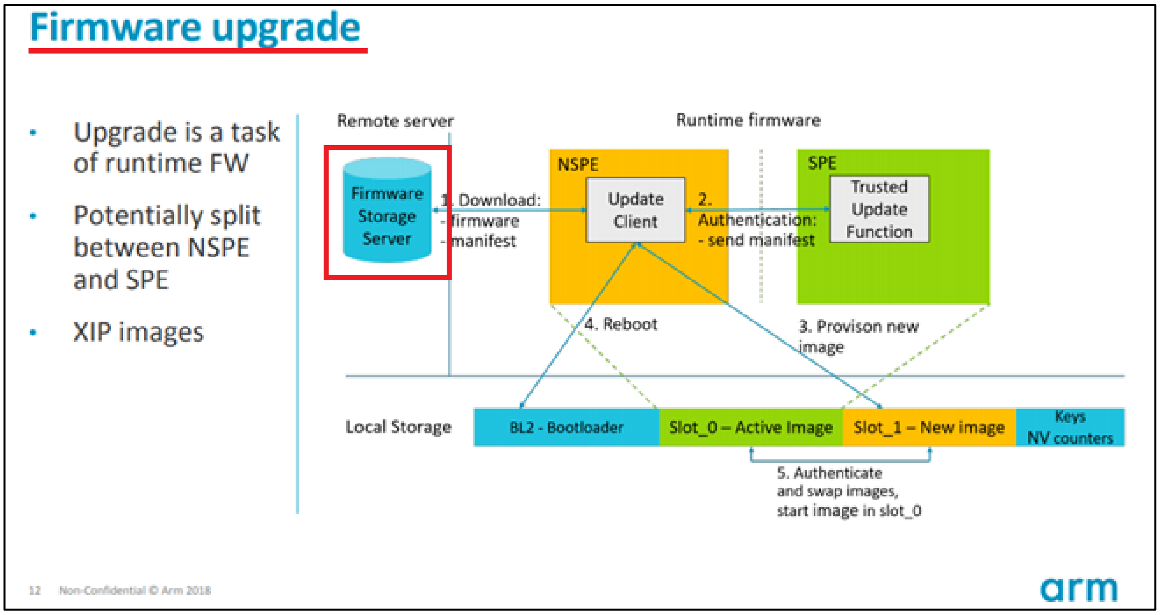


8 Update process (informational)

Some devices require protection against failure of a new image by retention of a known good image, normally the current image. This implies sufficient NVM to store two images. The simplest case is when both images might be stored on the device in eFlash, in which case the eFlash has to be dimensioned for two image slots, a primary slot and a secondary slot. The same principle can be applied for external flash. Following the download and processing of a new image the update client of the device is responsible for programming the new image into the secondary slot and arranging for the device to be rebooted. Images that have been provisioned to storage are known as candidate images.

The update process might fetch images from an external interface such as USB, UART, SD-MMC, NAND, NOR, Ethernet to SoC NVM memories such as NAND Flash, LPPDR2-NVM or any memory as indicated by SoC inputs pins.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Page 42).



Firmware upgrade

- Upgrade is a task of runtime FW
- Potentially split between NSPE and SPE
- XIP images

12 Non-Confidential © Arm 2018

arm

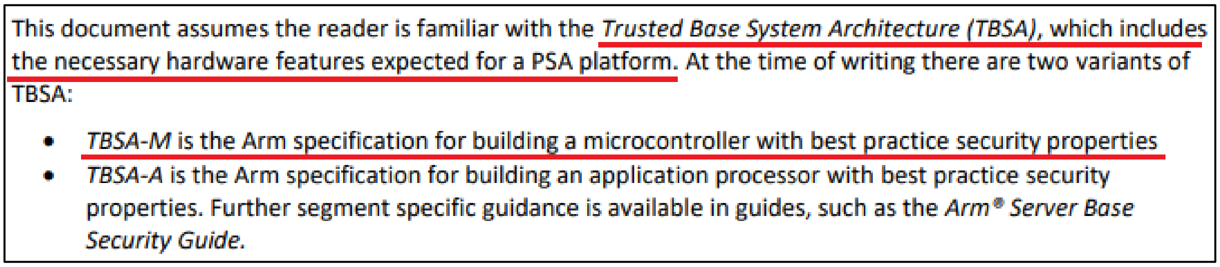
Source: http://connect.linaro.org.s3.amazonaws.com/hkg18/presentations/hkg18-223.pdf (Slide 12).

62. The PAN1780’s boot code 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.

63. For example, the device supports a secure boot flow process that includes an anti-rollback function. The anti-roll back function ensures that latest version of the firmware is executed by preventing the execution of older versions.

64. During the secure boot flow process after the device is powered on, the Immutable section of the firmware (“boot code”) is executed first. The next section of firmware is executed *only after* the Immutable section of the firmware (“boot code”) validates and verifies the version of the Updateable section of the firmware (“system code”). During the boot flow process, the micro-controller’s processor executes the “Update Client” and a “Trusted Update Function” (“update agent interface programming”) to retrieve the firmware update (“second

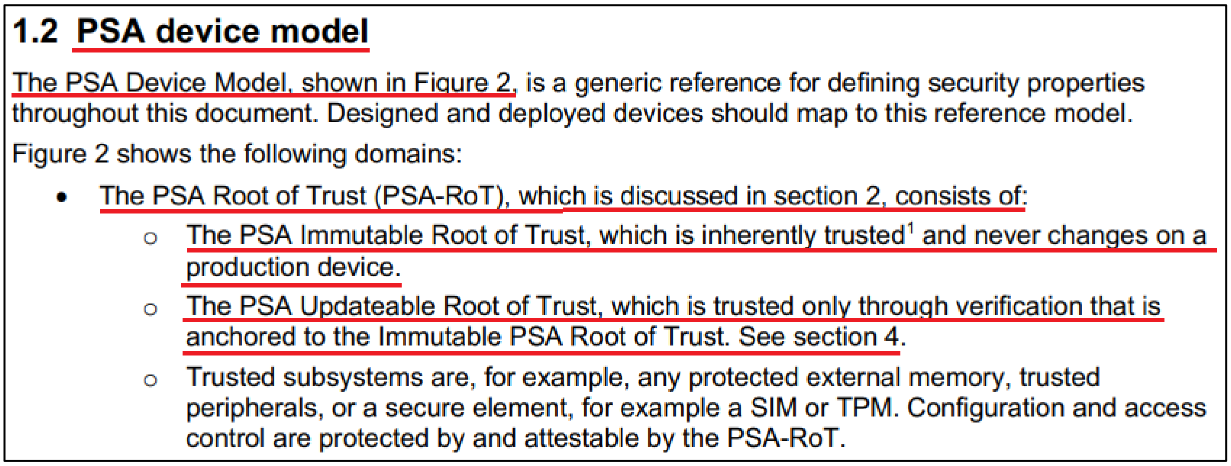
system software") from external server or device ("external data storage device") through a connecting interface, such as USB, UART, SD-MMC, or Ethernet.



This document assumes the reader is familiar with the *Trusted Base System Architecture (TBSA)*, which includes the necessary hardware features expected for a PSA platform. At the time of writing there are two variants of TBSA:

- *TBSA-M* is the Arm specification for building a microcontroller with best practice security properties
- *TBSA-A* is the Arm specification for building an application processor with best practice security properties. Further segment specific guidance is available in guides, such as the *Arm® Server Base Security Guide.*

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Pages 13 and 14).



1.2 PSA device model

The PSA Device Model, shown in Figure 2, is a generic reference for defining security properties throughout this document. Designed and deployed devices should map to this reference model.

Figure 2 shows the following domains:

- The PSA Root of Trust (PSA-RoT), which is discussed in section 2, consists of:
  - The PSA Immutable Root of Trust, which is inherently trusted[1] and never changes on a production device.
  - The PSA Updateable Root of Trust, which is trusted only through verification that is anchored to the Immutable PSA Root of Trust. See section 4.
  - Trusted subsystems are, for example, any protected external memory, trusted peripherals, or a secure element, for example a SIM or TPM. Configuration and access control are protected by and attestable by the PSA-RoT.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 4).

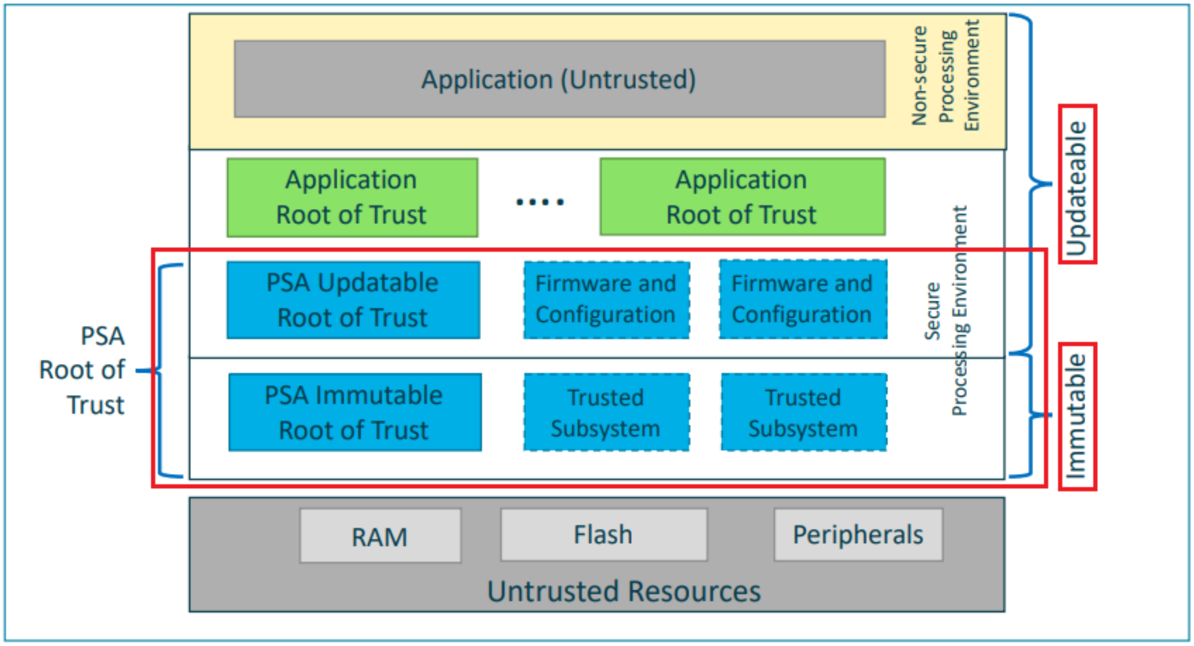
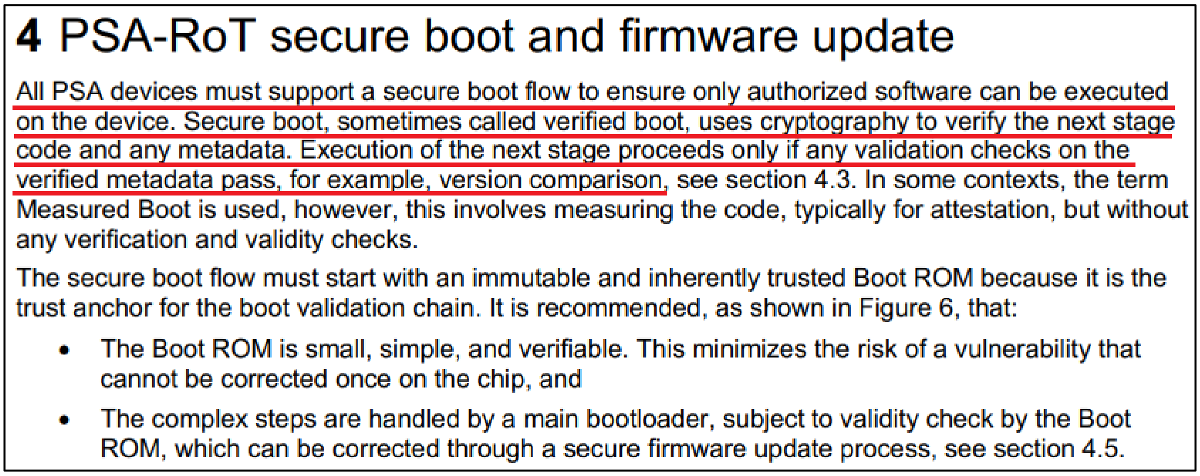

Application (Untrusted)
Non-secure Processing Environment
Application Root of Trust
....
Application Root of Trust
PSA Updatable Root of Trust
Firmware and Configuration
Firmware and Configuration
Secure Processing Environment
PSA Root of Trust
PSA Immutable Root of Trust
Trusted Subsystem
Trusted Subsystem
Updateable
Immutable
RAM
Flash
Peripherals
Untrusted Resources

Figure 2: PSA device model

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 5).



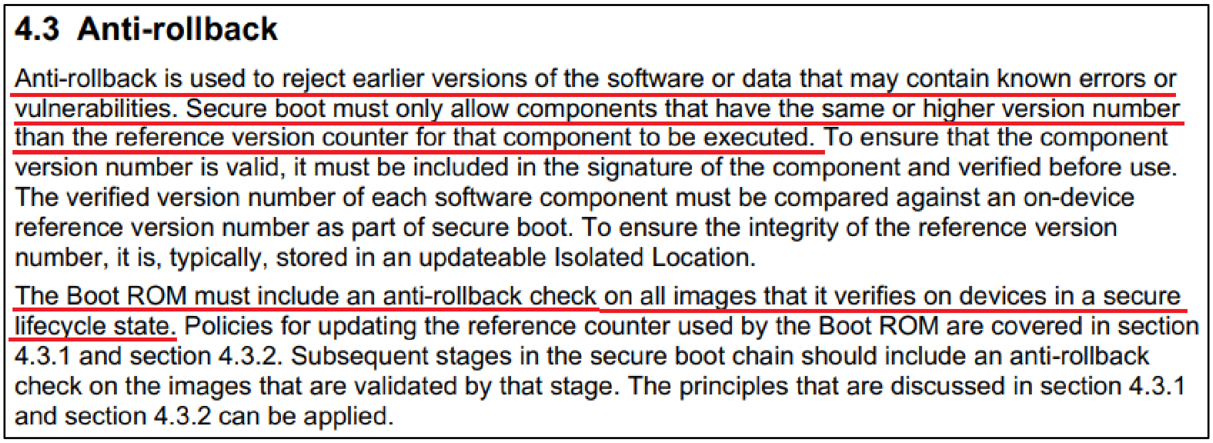
4 PSA-RoT secure boot and firmware update

All PSA devices must support a secure boot flow to ensure only authorized software can be executed on the device. Secure boot, sometimes called verified boot, uses cryptography to verify the next stage code and any metadata. Execution of the next stage proceeds only if any validation checks on the verified metadata pass, for example, version comparison, see section 4.3. In some contexts, the term Measured Boot is used, however, this involves measuring the code, typically for attestation, but without any verification and validity checks.

The secure boot flow must start with an immutable and inherently trusted Boot ROM because it is the trust anchor for the boot validation chain. It is recommended, as shown in Figure 6, that:

- The Boot ROM is small, simple, and verifiable. This minimizes the risk of a vulnerability that cannot be corrected once on the chip, and
- The complex steps are handled by a main bootloader, subject to validity check by the Boot ROM, which can be corrected through a secure firmware update process, see section 4.5.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 14).

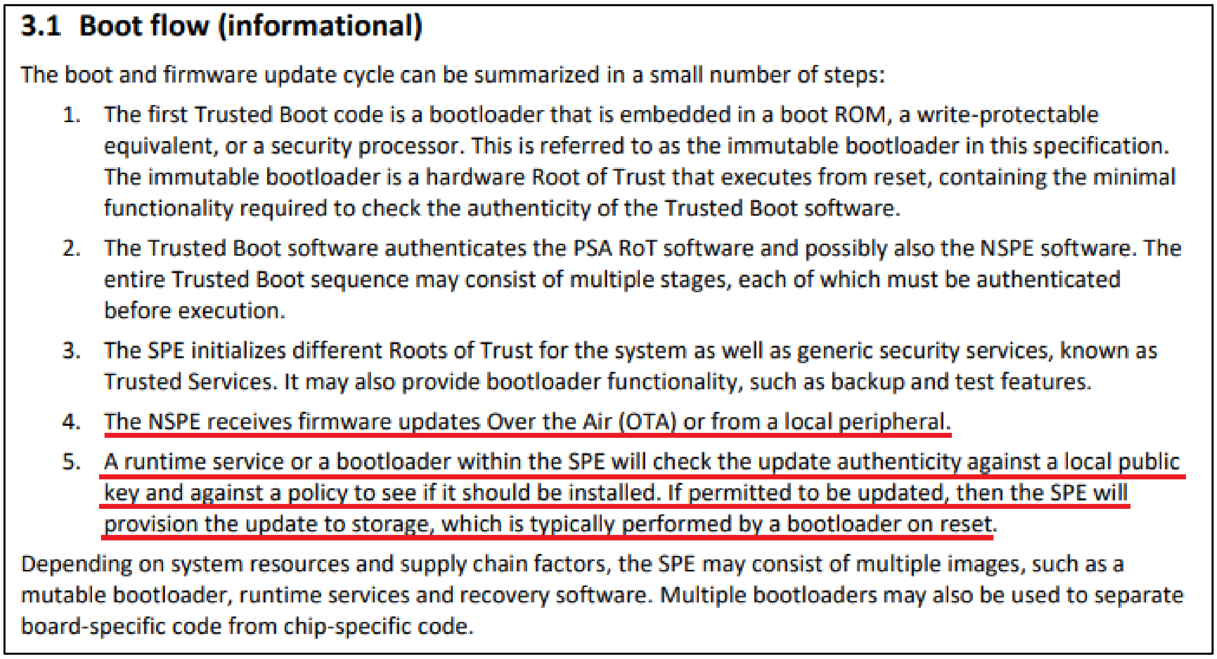


4.3 Anti-rollback

Anti-rollback is used to reject earlier versions of the software or data that may contain known errors or vulnerabilities. Secure boot must only allow components that have the same or higher version number than the reference version counter for that component to be executed. To ensure that the component version number is valid, it must be included in the signature of the component and verified before use. The verified version number of each software component must be compared against an on-device reference version number as part of secure boot. To ensure the integrity of the reference version number, it is, typically, stored in an updateable Isolated Location.

The Boot ROM must include an anti-rollback check on all images that it verifies on devices in a secure lifecycle state. Policies for updating the reference counter used by the Boot ROM are covered in section 4.3.1 and section 4.3.2. Subsequent stages in the secure boot chain should include an anti-rollback check on the images that are validated by that stage. The principles that are discussed in section 4.3.1 and section 4.3.2 can be applied.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0083_PSA_TBSA-M_1.0-bet2.pdf?revision=95776bd7-b790-48f0-bb18-ee064fb381ad (Page 16).



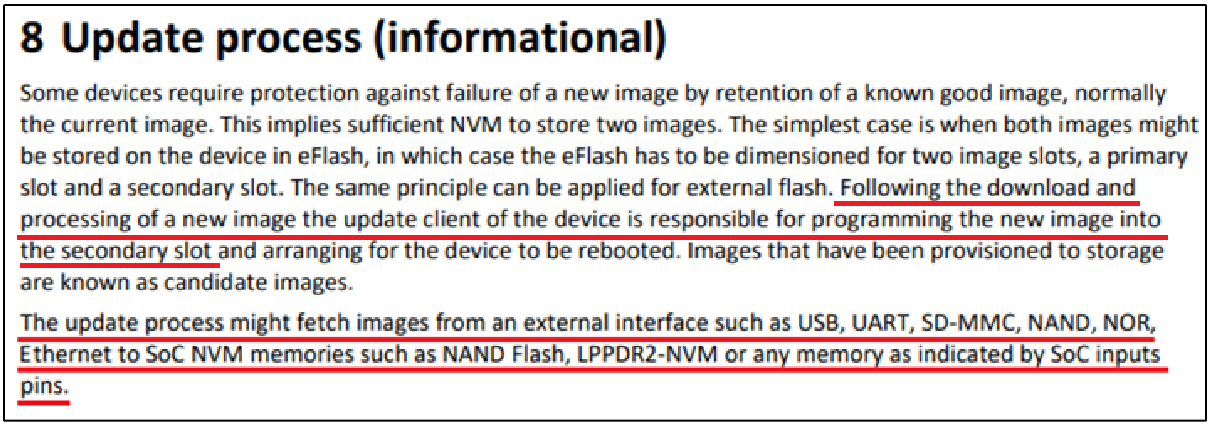
3.1 Boot flow (informational)

The boot and firmware update cycle can be summarized in a small number of steps:

1. The first Trusted Boot code is a bootloader that is embedded in a boot ROM, a write-protectable equivalent, or a security processor. This is referred to as the immutable bootloader in this specification. The immutable bootloader is a hardware Root of Trust that executes from reset, containing the minimal functionality required to check the authenticity of the Trusted Boot software.
2. The Trusted Boot software authenticates the PSA RoT software and possibly also the NSPE software. The entire Trusted Boot sequence may consist of multiple stages, each of which must be authenticated before execution.
3. The SPE initializes different Roots of Trust for the system as well as generic security services, known as Trusted Services. It may also provide bootloader functionality, such as backup and test features.
4. The NSPE receives firmware updates Over the Air (OTA) or from a local peripheral.
5. A runtime service or a bootloader within the SPE will check the update authenticity against a local public key and against a policy to see if it should be installed. If permitted to be updated, then the SPE will provision the update to storage, which is typically performed by a bootloader on reset.

Depending on system resources and supply chain factors, the SPE may consist of multiple images, such as a mutable bootloader, runtime services and recovery software. Multiple bootloaders may also be used to separate board-specific code from chip-specific code.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Page 18).

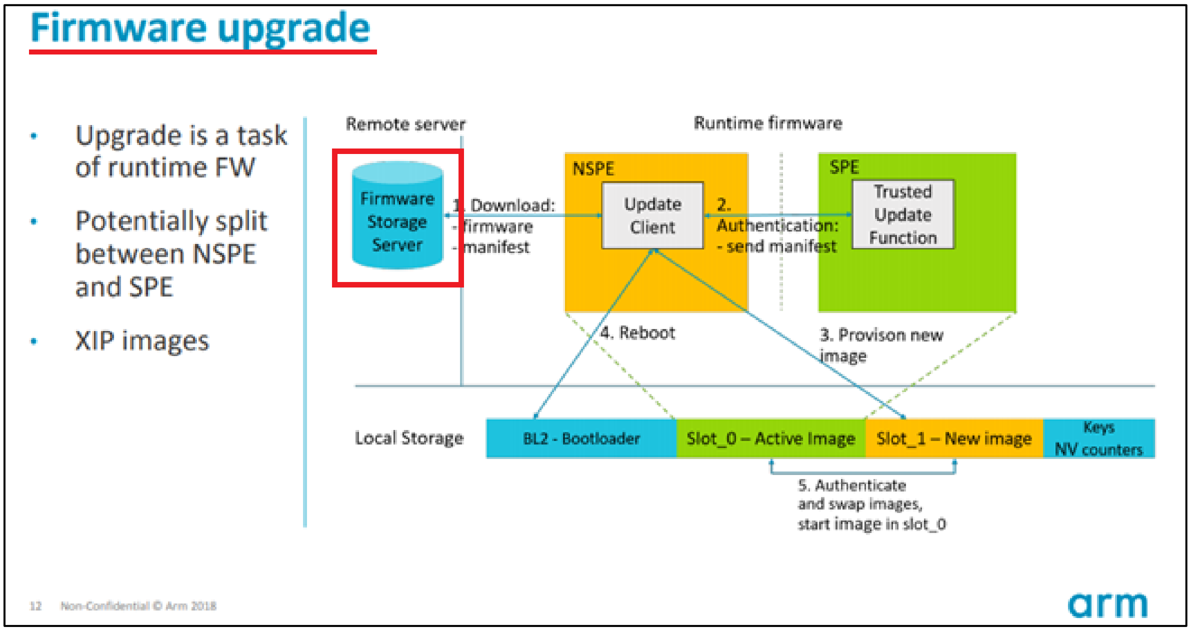


8 Update process (informational)

Some devices require protection against failure of a new image by retention of a known good image, normally the current image. This implies sufficient NVM to store two images. The simplest case is when both images might be stored on the device in eFlash, in which case the eFlash has to be dimensioned for two image slots, a primary slot and a secondary slot. The same principle can be applied for external flash. Following the download and processing of a new image the update client of the device is responsible for programming the new image into the secondary slot and arranging for the device to be rebooted. Images that have been provisioned to storage are known as candidate images.

The update process might fetch images from an external interface such as USB, UART, SD-MMC, NAND, NOR, Ethernet to SoC NVM memories such as NAND Flash, LPPDR2-NVM or any memory as indicated by SoC inputs pins.

Source: https://developer.arm.com/-/media/Arm%20Developer%20Community/PDF/PSA/DEN0072-PSA_TBFU_1.1-BETA0.pdf?revision=3ce2513a-ae0f-4b43-96a0-851ed67a640b (Page 42).



Firmware upgrade

- Upgrade is a task of runtime FW
- Potentially split between NSPE and SPE
- XIP images

12 Non-Confidential © Arm 2018

Source: http://connect.linaro.org.s3.amazonaws.com/hkg18/presentations/hkg18-223.pdf (Slide 12).

65. Panasonic has had knowledge of the ’612 Patent at least as of the date when it was notified of the filing of this action.

66. Liberty Patents has been damaged as a result of the infringing conduct by Panasonic alleged above. Thus, Panasonic 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.

67. 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

68. Panasonic has also indirectly infringed the ’959 Patent, the ’573 Patent, and the ’612 Patent by inducing others to directly infringe the ’959 Patent, the ’573 Patent, and the ’612 Patent. Panasonic has induced the end-users, Panasonic’s customers, to directly infringe (literally and/or under the doctrine of equivalents) the ’959 Patent, the ’573 Patent, and the ’612 Patent by using the accused products.

69. Panasonic 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 1 of the ’959 Patent, claim 13 of the ’573 Patent, and claim 1 of the ’612 Patent.

70. Such steps by Panasonic 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.

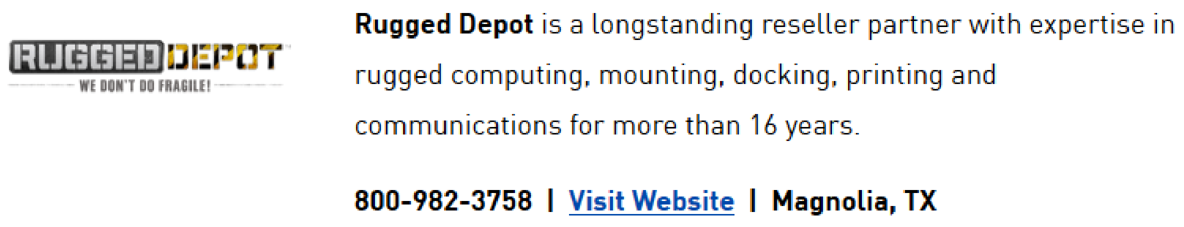
71. Panasonic performed these steps, which constitute induced infringement, with the knowledge of the ’959 Patent, the ’573 Patent, and the ’612 Patent and with the knowledge that the induced acts constitute infringement.

72. Panasonic was and is aware that the normal and customary use of the accused products by Panasonic’s customers would infringe the ’959 Patent, the ’573 Patent, and the ’612 Patent. Panasonic’s inducement is ongoing.

73. Panasonic has also induced its affiliates, or third-party manufacturers, shippers, distributors, retailers, or other persons acting on its or its affiliates’ behalf, to directly infringe (literally and/or under the doctrine of equivalents) the ’959 Patent, the ’573 Patent, and the ’612 Patent by importing, selling or offering to sell the accused products, including, for example, Amazon, CDW, OC Rugged Laptops, Walmart, and others.

74. Panasonic 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.

75. Panasonic 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. Panasonic states, for example, that it sells TOUGHBOOK® laptops through a network of authorized reseller partners. One of these partners is Rugged Depot, a company located in 27060 Decker Prairie Rosehill Road, Magnolia, Texas, 77355.

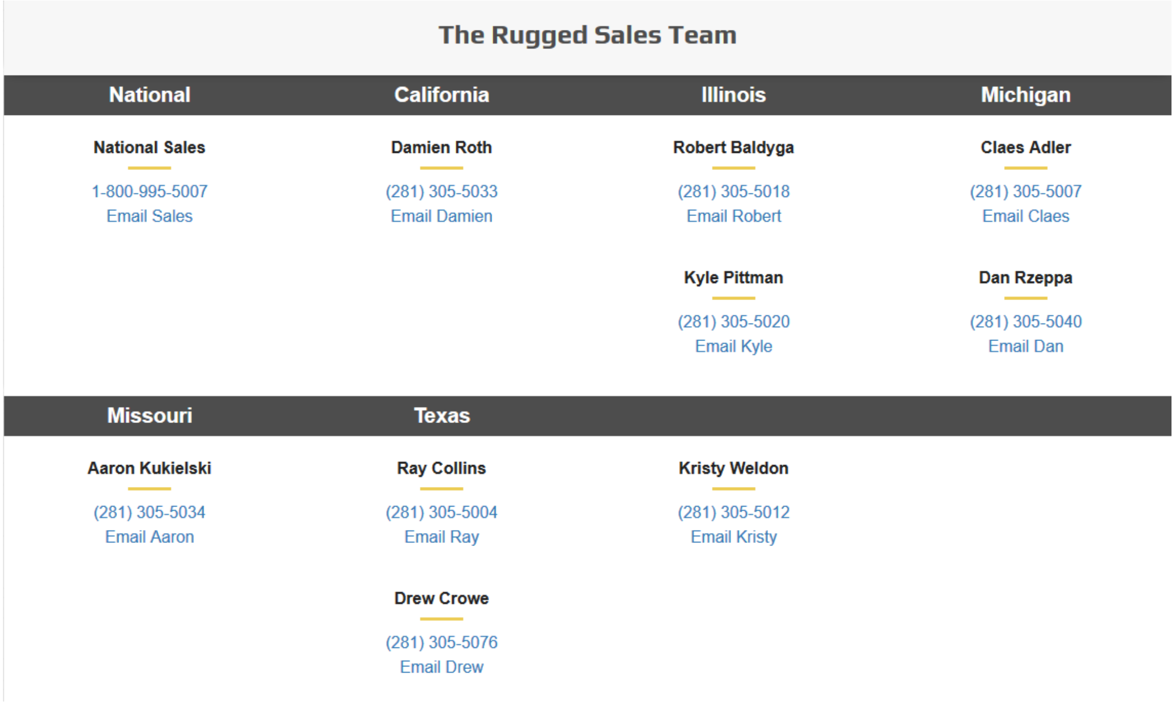


Rugged Depot is a longstanding reseller partner with expertise in rugged computing, mounting, docking, printing and communications for more than 16 years.

800-982-3758 | Visit Website | Magnolia, TX

Source: https://na.panasonic.com/us/featured-reseller-partners-0

76. Rugged Depot has at least two members of its sales team devoted to sales in Texas:



The Rugged Sales Team

National	California	Illinois	Michigan
National Sales 1-800-995-5007 Email Sales	Damien Roth (281) 305-5033 Email Damien	Robert Baldyga (281) 305-5018 Email Robert	Claes Adler (281) 305-5007 Email Claes
		Kyle Pittman (281) 305-5020 Email Kyle	Dan Rzeppa (281) 305-5040 Email Dan
Missouri	**Texas**		
Aaron Kukielski (281) 305-5034 Email Aaron	Ray Collins (281) 305-5004 Email Ray	Kristy Weldon (281) 305-5012 Email Kristy	
	Drew Crowe (281) 305-5076 Email Drew		

Source: https://ruggeddepot.com/contact/

77. Panasonic purposefully directs or controls the sale of the accused products into established United States distribution channels, including sales to nationwide retailers. Panasonic's established United States distribution channels include one or more United States based affiliates.

78. Panasonic purposefully 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.

79. Panasonic purposefully places the accused products—whether by itself or through subsidiaries—into an international supply chain, knowing that the accused products will be sold

in the United States, including Texas. Therefore, Panasonic also facilitates the sale of the accused products in Texas.

80. Panasonic 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 1 of the ’959 Patent, claim 13 of the ’573 Patent, and claim 1 of the ’612 Patent. For example, Panasonic has established a contract with the State of Texas to provide Panasonic Branded Manufacturer Hardware, including Panasonic Toughbooks. *See* https://info.panasonic.com/Texas-Contract.html.

81. Such steps by Panasonic 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.

82. Panasonic performed these steps, which constitute induced infringement, with the knowledge of the ’959 Patent, the ’573 Patent, and the ’612 Patent and with the knowledge that the induced acts would constitute infringement.

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

84. Panasonic’s inducement is ongoing.

85. Panasonic has also indirectly infringed by contributing to the infringement of the ’959 Patent, the ’573 Patent, and the ’612 Patent. Panasonic has contributed to the direct

infringement of the ’959 Patent, the ’573 Patent, and the ’612 Patent by the end-user of the accused products.

86. 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 ’959 Patent, the ’573 Patent, and the ’612 Patent, for example, claim 1 of the ’959 Patent, claim 13 of the ’573 Patent, and claim 1 of the ’612 Patent.

87. The special features include, for example, executing computer instructions in an instruction cache used in a manner that infringes the ’959 Patent; 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.

88. These special features constitute a material part of the invention of one or more of the claims of the ’959 Patent, the ’573 Patent, and the ’612 Patent and are not staple articles of commerce suitable for substantial non-infringing use.

89. Panasonic’s contributory infringement is ongoing.

90. Panasonic has had actual knowledge of the ’959 Patent, 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, Panasonic has known the scope of the claims of the ’959 Patent, the ’573 Patent, and the ’612 Patent; the products that practice the ’959 Patent, the ’573 Patent, and the ’612 Patent; and that Liberty Patents is the owner of the ’959 Patent, the ’573 Patent, and the ’612 Patent.

91. By the time of trial, Panasonic 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 ’959 Patent, the ’573 Patent, and the ’612 Patent.

92. Furthermore, Panasonic 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).

93. Panasonic'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 Panasonic. Panasonic has knowledge of the '959 Patent, the '573 Patent, and the '612 Patent.

94. Panasonic's customers have infringed the '959 Patent, the '573 Patent, and the '612 Patent. Panasonic has encouraged its customers' infringement.

95. Panasonic's direct and indirect infringement of the '959 Patent, 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.

96. Liberty Patents has been damaged as a result of Panasonic's infringing conduct alleged above. Thus, Panasonic 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 Panasonic, and that the Court grant Liberty Patents the following relief:

a. Judgment that one or more claims of the '959 Patent, the '573 Patent, and the '612 Patent have been infringed, either literally and/or under the doctrine of equivalents, by

Panasonic and/or all others acting in concert therewith;

b. A permanent injunction enjoining Panasonic and its officers, directors, agents, servants, affiliates, employees, divisions, branches, subsidiaries, parents, and all others acting in concert therewith from infringement of the ’959 Patent, the ’573 Patent, and the ’612 Patent; or, in the alternative, an award of a reasonable ongoing royalty for future infringement of the ’959 Patent, the ’573 Patent, and the ’612 Patent by such entities;

c. Judgment that Panasonic account for and pay to Liberty Patents all damages to and costs incurred by Liberty Patents because of Panasonic’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 Panasonic’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: September 2, 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