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

ADVANCED CODING TECHNOLOGIES LLC,	§	Case No.
	§	**JURY TRIAL DEMANDED**
Plaintiff,	§	
v.	§	
APPLE INC.,	§	
Defendant.	§	

COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff Advanced Coding Technologies LLC ("ACT" or "Plaintiff"), for its Complaint against Defendant Apple Inc. (collectively "Apple" or "Defendant") for patent infringement under 35 U.S.C. § 271, alleges as follows:

THE PARTIES

1. ACT is a limited liability company organized and existing under the laws of the State of Texas, with its principal place of business located at 104 East Houston Street, Suite 140, Marshall, Texas 75670.

2. Defendant Apple is a corporation organized and existing under the laws of California. Apple is one of the leading mobile phone, handset, and PC sellers in the United States and the world. Apple sells its products directly to consumers at physical Best Buy locations. There are multiple Best Buy locations in this District where Apple products are sold directly to customers, including at least: 823 North Creek Drive, Sherman, Texas 75092 (Grayson County), 2800 North Central Expressway, Plano, Texas 75074 (Collin County); 1751 North Central Expressway, Suite C, McKinney, Texas 75070 (Collin County); 3333 Preston Road, Suite 200, Frisco, Texas 75034

(Denton County); 5299 Eldorado Parkway, Frisco, Texas 75033 (Denton County); 1800 South Loop 288, Suite 102 Building 1; Denton, Texas 76205 (Denton County); 6060 Long Prairie Road, Suite 500, Flower Mound, Texas 75028 (Denton County); 2601 South Stemmons Freeway, Suite 300, Lewisville, Texas 75067 (Denton County); 5885 East Freeway, Beaumont, Texas 77706 (Jefferson County); 8725 Memorial Boulevard, Port Arthur, Texas 77640 (Jefferson County); 869 NE Mall Boulevard, Hurst, Texas 76053 (Shelby County); 422 West Loop 281, Suite 100, Longview, Texas 75605 (Gregg County); 4210 Saint Michael Drive, Texarkana, Texas 75503 (Bowie County); 5514 South Broadway Avenue, Tyler, Texas 75703 (Smith County). https://stores.bestbuy.com/tx/sherman/823-n-creek-dr-1023.html#shop; https://stores.bestbuy.com/tx/plano/2800-n-central-expy-202.html; https://stores.bestbuy.com/tx/frisco/3333-preston-rd-180.html; https://stores.bestbuy.com/tx/frisco/5299-eldorado-pkwy-1773.html; https://stores.bestbuy.com/tx/beaumont/5885-eastex-fwy-238.html; https://stores.bestbuy.com/tx/denton/1800-s-loop-288-827.html; https://stores.bestbuy.com/tx/flower-mound/6060-long-prairie-rd-1038.html; https://stores.bestbuy.com/tx/lewisville/2601-s-stemmons-fwy-258.html; https://stores.bestbuy.com/tx/longview/422-w-loop-281-594.html; https://stores.bestbuy.com/tx/mckinney/1751-n-central-expy-196.html; https://stores.bestbuy.com/tx/port-arthur/8725-memorial-blvd-1545.html; https://stores.bestbuy.com/tx/texarkana/4210-saint-michael-dr-605.html; https://stores.bestbuy.com/tx/tyler/5514-s-broadway-ave-246.html

3. Further, certain Best Buy locations within this District contain Apple Shops. According to Apple's website, "Apple Shops are Apple-designed outlets located within select

Apple resellers and other retail stores. Many are staffed with Apple Solutions Consultants — trained Apple employees who can help you find the best solution." *See* https://locate.apple.com/sales?pt=6&lat=33.021827697753906&lon=-96.69925689697266&address=Plano%2C+TX. Apple advertises on its website that the Best Buy located at 2800 North Central Expressway, Plano, Texas 75074-5415 (Collin County), contains an Apple Shop. *See* https://locate.apple.com/sales?pt=6&lat=33.021827697753906&lon=-96.69925689697266&address=Plano%2C+TX. Apple further advertises on its website other Best Buy locations within this District that contain Apple Shops, including the store located at 190 East Stacy Road, Allen, Texas 75002-8734 (Collin County). *See* https://locate.apple.com/sales?pt=6&lat=33.021827697753906&lon=-96.69925689697266&address=Plano%2C+TX.

Find Locations

Sales

Apple Stores and Apple Authorized Resellers

‹ Back to Home

Enter Location, City or Zipcode
Plano, TX

Select a product
Apple Watch
Change

Reset | Go

99 Apple Watch sales locations near **Plano, TX.**

Filters

1 VERIZON - COR - 7092801
741 N CENTRAL EXPY
PLANO, TX, 75075
iPhone iPad Watch Apple TV 0.9 mi

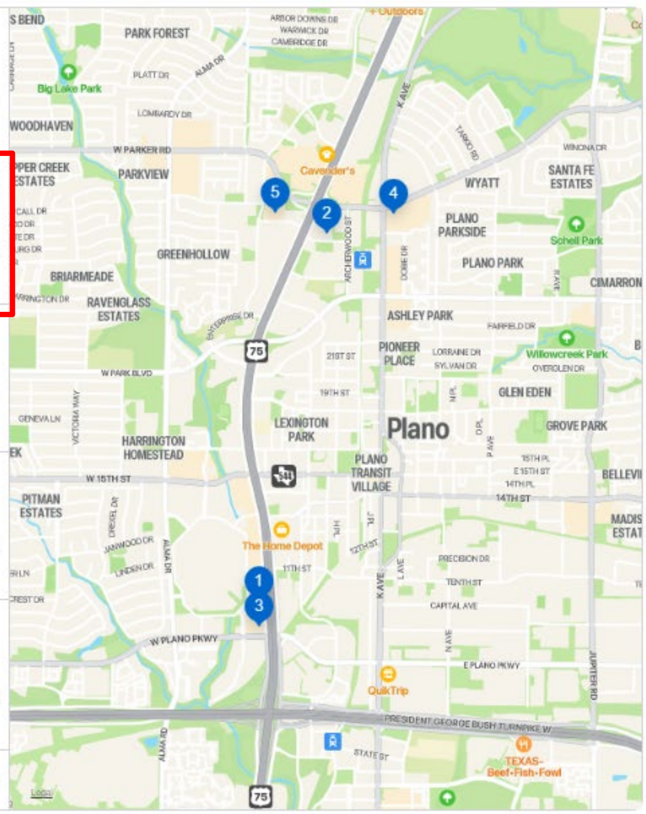
2 BEST BUY - 0202 APPLE SHOP
2800 N CENTRAL EXPY, PLANO TX
PLANO, TX, 75074-5415
iPhone Mac iPad Watch Apple TV HomePod iPod 1.0 mi

3 AT&T - COR - 0091901
701 N CENTRAL EXPY STE 400
PLANO, TX, 75075
iPhone iPad Watch 1.0 mi

4 T-MOBILE - COR - 8661
1110 PARKER ROAD EAST SUITE C
PLANO, TX, 75074
iPhone iPad Watch Apple TV 1.0 mi

5 TARGET STORE - 0067
120 W PARKER RD
PLANO, TX, 75075-2331
iPhone iPad Watch Apple TV HomePod iPod 1.2 mi

Next ›

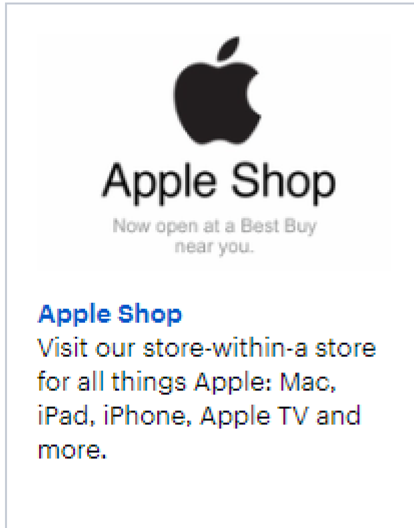


APPLE SHOPS

Apple Shops are Apple-designed outlets located within select Apple resellers and other retail stores. Many are staffed with Apple Solutions Consultants — trained Apple employees who can help you find the best solution.

https://locate.apple.com/sales?pt=6&lat=33.021827697753906&lon=-96.69925689697266&address=Plano%2C+TX

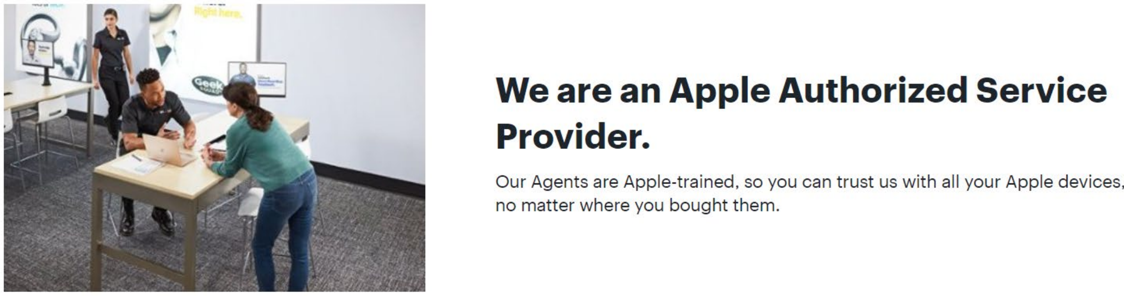
4. By way of further example, Best Buy also advertises on its website that the location in Plano, Texas, is an Apple Shop.



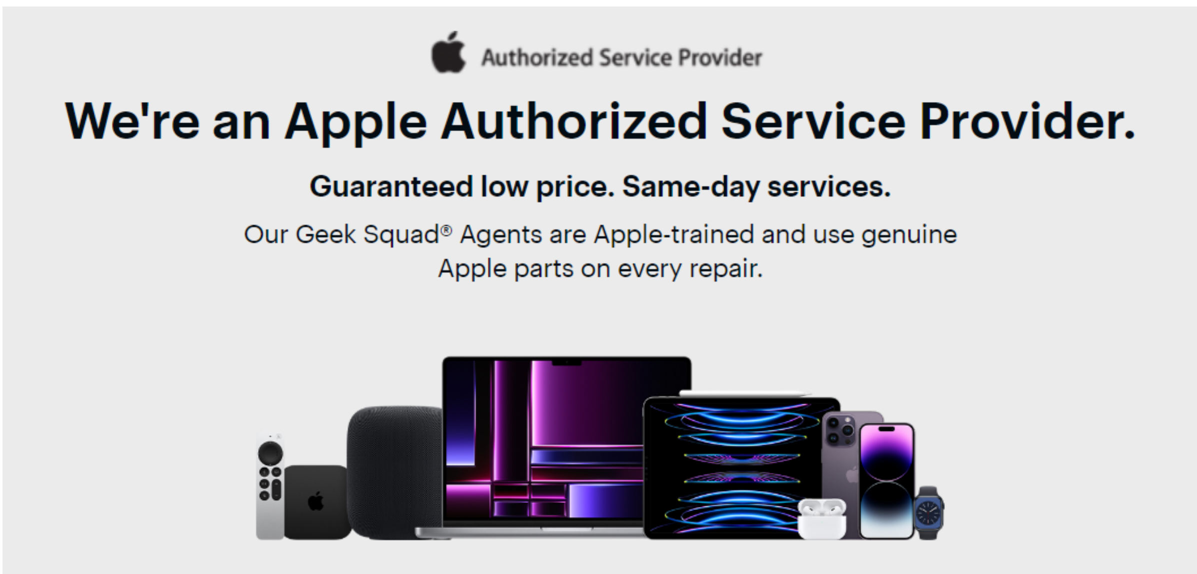
https://stores.bestbuy.com/tx/plano/2800-n-central-expy-202.html

5. Best Buy also advertises on its website that an Apple Shop is located at the store in Allen, Texas. https://stores.bestbuy.com/tx/allen/190-e-stacy-rd-1780.html

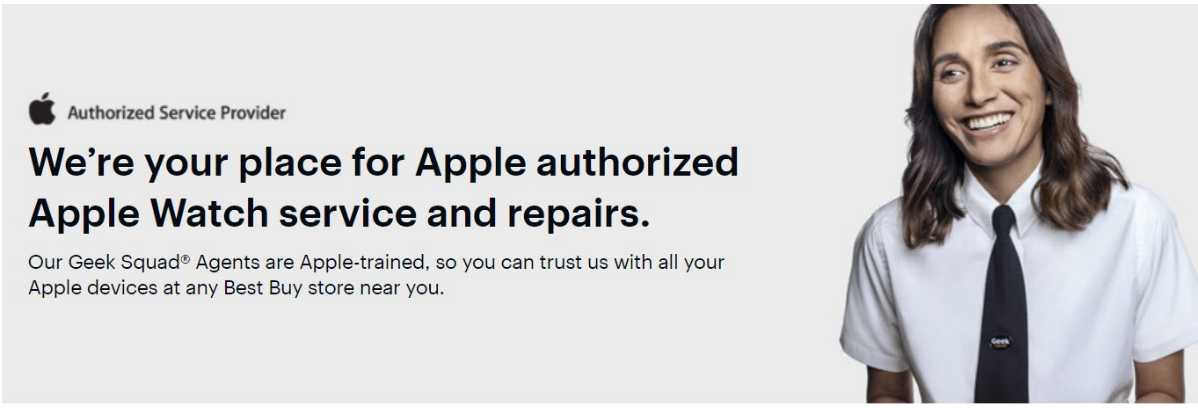
6. Further, certain Best Buy locations, including those located in this District, are "Apple Authorized Service Providers" and the "Geek Squad Agents" at Best Buy locations are "Apple-trained and use genuine Apple parts on every repair." https://www.bestbuy.com/site/services/apple-service repair/pcmcat1554741516170.c?id=pcmcat1554741516170



https://www.bestbuy.com/site/apple-service-repair/apple-watch-service-repair/pcmcat1554832617549.c?id=pcmcat1554832617549



https://www.bestbuy.com/site/services/apple-service-repair/pcmcat1554741516170.c?id=pcmcat1554741516170



Apple Watch repairs at Best Buy.

Apple Watch battery replacement and screen repair.

If you're searching for "Apple Watch screen repair near me," look no further than your local Best Buy. Even though Apple Watch screens are designed with durable, scratch-resistant materials and have water-resistant capabilities of various depths depending on the model, accidents can happen. No matter where you purchased your Apple Watch, you can trust our Apple-trained technicians with all your Apple Watch repairs. As a full-service Apple Authorized Service Provider, we only use genuine Apple parts that meet high standards so that the repair to your watch is backed by Apple. Our highly skilled Geek Squad® Agents can fix Apple Watch screens and address problems with the speaker or mic; they can also repair your AirPods since you might be having trouble using them with your watch. Plus, we can perform Wi-Fi troubleshooting, diagnose software issues, help with operating system upgrades and take care of Apple Watch battery replacements. Whether you have the latest Apple Watch or an older model, bring in your cracked screen and our technicians trained in Apple Watch glass repair can make it as good as new with an Apple Watch screen replacement.

Scheduling your Apple Watch screen repair.

You may be wondering how Best Buy's repair fees compare to other service providers or if your Apple Watch repairs are covered under warranty. Rest assured that our Apple Watch repair costs match those of the Apple store. Also, while every Apple Watch comes with the Apple Limited Warranty for manufacturing defects, that warranty does vary by model. You can extend coverage on your watch for two to three additional years with the purchase of AppleCare+ for Apple Watch, which includes coverage for up to two incidents of accidental damage and technical support over the phone. You can add this extra protection when you purchase your Apple Watch or for a limited time thereafter. Additionally, you can save on the cost of Apple Watch repairs when you become a My Best Buy Total™ member.

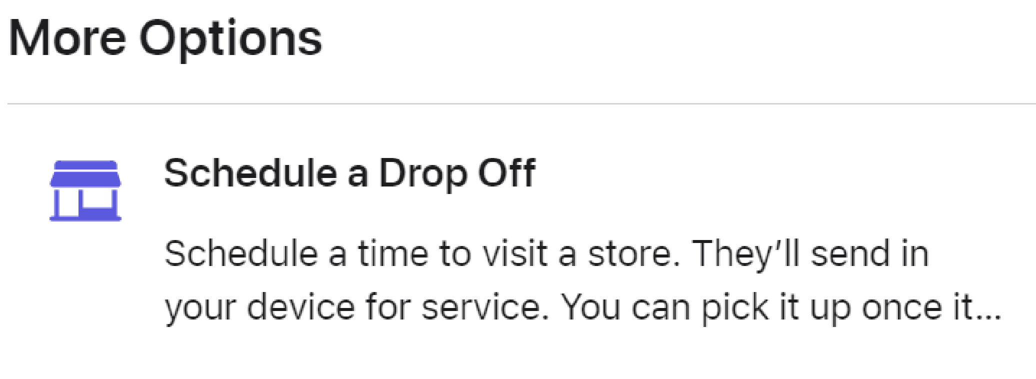
Before you come in for an Apple Watch repair, first try to restart the device to see if that helps solve the problem you were experiencing. Another quick fix that sometimes works is unpairing your Apple Watch from your smartphone and then re-pairing it. If those steps don't provide positive results, reserve an appointment at a nearby Best Buy location. You can expedite your visit by making sure that your watch is fully charged and unpaired from your iPhone. Also, remove your Apple Watch band and store it with your charger for use once your repair is complete. If your other devices need attention, you can schedule Apple iPad repairs and iPhone repairs at the same time for added convenience. Be sure you know your Apple ID and password before you arrive to help make the process even smoother.

https://www.bestbuy.com/site/apple-service-repair/apple-watch-service-repair/pcmcat1554832617549.c?id=pcmcat1554832617549

7. Upon information and belief, Defendant employs individuals in this Judicial District involved in the technology, sales, and marketing of its products.

8. Apple touted the expansion of its "Apple authorized service network" at "nearly 1,000 Best Buy stores across the U.S. now providing expert service and repairs for Apple products." https://www.apple.com/newsroom/2019/06/apple-partners-with-best-buy-for-expanded-repair-service/. Further, Apple states that "Apple-certified repairs at an Apple store or

an authorized service provider are performed by trained experts who use genuine Apple parts. Every repair is backed by Apple." https://www.apple.com/newsroom/2019/06/apple-partners-with-best-buy-for-expanded-repair-service/. When a customer or user seeks to schedule a repair from Apple's website, Apple directs its users to Best Buy locations within this District.

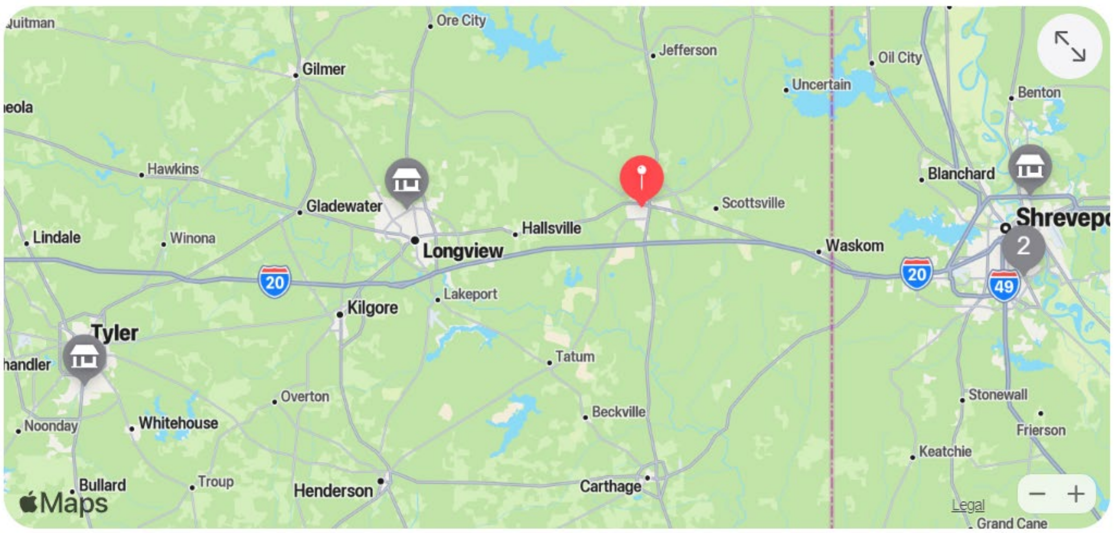


Schedule a visit.

Choose a location

Showing options near Marshall, TX.

Appointments available



Best Buy - Longview
23.1 mi • 422 W Loop 281, Longview
Available tomorrow, 12:20 PM

https://getsupport.apple.com/solutions

9. Apple advertises on its website these Best Buy locations as Apple Authorized Service Providers as certified "so you get the same professionalism and quality of repair you'd expect from Apple." https://getsupport.apple.com/solutions/schedule-repair/providers

10. Apple has admitted that, "[w]hile at Best Buy, Apple's employees provide information about Apple's products to prospective customers" and that "[m]any are staffed with

Apple Solutions Consultants – trained Apple employees who can help you find the best solution.”). *Slyde Analytics LLC, v. Apple Inc.*, Case No. 2:24-cv-00331-RWS-RSP, Dkt. 21, 11 (citations omitted). Upon information and belief, Defendant employs individuals in this Judicial District involved in the technology, sales, and marketing of its products.

JURISDICTION

11. This is an action for patent infringement arising under the patent laws of the United States, 35 U.S.C. §§ 1, *et seq*. This Court has jurisdiction over this action pursuant to 28 U.S.C. §§ 1331 and 1338(a).

12. This Court has specific and personal jurisdiction over Defendant consistent with the requirements of the Due Process Clause of the United States Constitution and the Texas Long Arm Statute. Upon information and belief, Defendant has sufficient minimum contacts with the forum because Defendant transacts substantial business in the State of Texas and in this Judicial District. Further, Defendant has, directly or through subsidiaries or intermediaries, committed and continues to commit acts of patent infringement in the State of Texas and in this Judicial District as alleged in this Complaint, alleged more particularly below.

13. Venue is proper in this Judicial District as to Defendant pursuant to 28 U.S.C. §§ 1400(b) and 1391(b) and (c) because Defendant is subject to personal jurisdiction in this Judicial District, has committed acts of patent infringement in this Judicial District, and has a regular and established place of business in this Judicial District. Defendant, through its own acts, makes, uses, sells, and/or offers to sell infringing products within this Judicial District, regularly does and solicits business in this Judicial District, and has the requisite minimum contacts with the Judicial District such that this venue is a fair and reasonable one.

PATENTS-IN-SUIT

14. On May 26, 2015, the United States Patent and Trademark Office duly and legally issued U.S. Patent No. 9,042,448 (the "'448 Patent") entitled "Moving Picture Encoding System, Moving Picture Encoding Method, Moving Picture Encoding Program, Moving Picture Decoding System, Moving Picture Decoding Method, Moving Picture Decoding Program, Moving Picture Reencoding System, Moving Picture Reencoding Method, and Moving Picture Reencoding Program." A true and correct copy of the '448 Patent is attached hereto as Exhibit A.

15. On July 24, 2012, the United States Patent and Trademark Office duly and legally issued U.S. Patent No. 8,230,101 (the "'101 Patent") entitled "Server Device for Media, Method for Controlling Server for Media, and Program." A true and correct copy of the '101 Patent is attached hereto as Exhibit B.

16. On September 28, 2010, the United States Patent and Trademark Office duly and legally issued U. S. Patent No. 7,804,891 (the "'891 Patent") entitled "Device and Method for Judging Communication Quality and Program Used for the Judgment." A true and correct copy of the '891 Patent is attached hereto as Exhibit C.

17. ACT is the sole and exclusive owner of all right, title, and interest in the '448 Patent, the '101 Patent, and the '891 Patent (collectively, the "Patents-in-Suit") and holds the exclusive right to take all actions necessary to enforce its rights to the Patents-in-Suit, including the filing of this patent infringement lawsuit. ACT also has the right to recover all damages for past, present, and future infringement of the Patents-in-Suit.

FACTUAL ALLEGATIONS

18. The Patents-in-Suit generally relate to systems and methods for coding and decoding data efficiently.

19. The ’448 Patent generally relates to hierarchical encoding that implements a process for super-resolution enlargement of video signals. The technology described in the ’448 Patent was developed by Satoru Sakazume of JVC Kenwood Corporation.

20. The ’101 Patent generally relates to a server for media capable of smoothly dealing with a large amount of digital contents. The technology described in the ’101 Patent was developed by Satoru Sekiguchi, Yoshio Sonoda, Isao Nakamura, Masamichi Furukawa, Yoshihia Mashita, Tomoaki Yoshida, and Masahito Watanabe of Kabushiki Kaisha Kenwood.

21. The ’891 Patent generally relates to judging communication quality in a communication system, and a program for causing a computer to execute the judgment. The technology described in the ’891 Patent was developed by Taichi Majima of Kabushiki Kaisha Kenwood.

22. In January 2018, Defendant joined the Alliance for Open Media (“AOM”), a non-profit industry consortium headquartered in Wakefield, Massachusetts, and formed to develop open, royalty-free technology for multimedia delivery. https://bitmovin.com/apple-joins-av1-codec-consortium; https://aomedia.org/about/. AOM is the creator of AV1. https://aomedia.org/about/.

23. Apple’s Director of Embedded Media, Krasimir Kolarov, is currently listed on AOM’s Steering Committee:

The AOMedia Steering Committee is responsible for the leadership and general management of AOMedia, including setting its mission, vision and objectives. The Steering Committee consists of the following representatives:

Member	**Organization**
John Elovson	Amazon
Krasimir Kolarov	Apple
Dale Mohlenhoff (*Finance*)	Cisco
Matt Frost (Chair)	Google
Iole Moccagatta	Intel
David Ronca	Meta
Steven Lees	Microsoft
Daniel Nazer	Mozilla
Anne Aaron (*Vice-Chair, Communications / Membership*)	Netflix
Frans Sijstermans	nVIDIA
Kwang Pyo Choi	Samsung
Shan Liu	Tencent

Source: https://aomedia.org/about/

Krasimir Kolarov

Director, Embedded Media at Apple Inc.

Menlo Park, California, United States · Contact Info

2K followers · 500+ connections

See your mutual connections

Join to view profile

Message

Apple Inc.

Stanford University

About

• Extensive engineering, technical business development and marketing knowledge and experience in the areas of Embedded Media, Wireless Communication, Security, Cable and Broadband, Information Technology, Industry-University relationships and programs.
• Expert technical knowledge and experience in a number of areas - signal processing (image and video), multicore, compression, robotics, haptics, artificial intelligence, mathematics.
• Extensive product shipping, algorithmic development and experience with: video and image quality; novel video compression technologies, software and hardware; HDR, WCG, color and display technologies; adaptive streaming (FPS), broadband and cable technology; embedded printing (AirPrint) and other next generation technologies.

Specialties: Engineering Management, Strategic Partnerships, Media Standards, setting up new technology start-up, inventing and developing innovative technology and IP, securing customers and financing.

Pioneered, standartized and lead industry adoption of media video and image technologies (CMAF, MIAF and WebVC) .
Chair MPEG Applications Formats Group and CTA WAVE CMAF-IF (Industry Forum) Group.
On the Steering Committee of CTA WAVE and the Board of AOM (Alliance for Open Media).

Source:
https://www.linkedin.com/in/krasimirkolarov?original_referer=https%3A%2F%2Fwww.google.com%2F

24. Defendant has infringed and continue to infringe one or more of the Patents-in-Suit by making, using, selling, offering to sell, and/or importing, and by actively inducing others to make, use, sell, offer to sell, and/or import products, including Defendant's iPhones, iPads, Macs, and Defendant's software and services, such as Safari, QuickTime, and Apple TV+, and chipsets thereof, that implement the AV1 technology claimed by the Patents-in-Suit, as described below.

25. Apple has had actual notice of the Patents-in-Suit from either related prior litigations accusing products with similar AV1 functionalities involving direct competitors of Defendant or from its knowledge of ACT's patent portfolio in *Advanced Coding Technologies, LLC v. Apple, Inc.*, Case No. 2:24-cv-00572-JRG (E.D. Tex. July 22, 2024).

26. ACT has at all times complied with the marking provisions of 35 U.S.C. § 287 with respect to the Asserted Patents.

COUNT I
(Infringement of the '448 Patent)

27. Paragraphs 1 through 26 are incorporated by reference as if fully set forth herein.

28. ACT has not licensed or otherwise authorized Defendant to make, use, offer for sale, sell, or import any products that embody the inventions of the '448 Patent.

29. Defendant has and continues to directly infringe the '448 Patent, either literally or under the doctrine of equivalents, without authority and in violation of 35 U.S.C. § 271, by making, using, offering to sell, selling, and/or importing into the United States products that satisfy each and every limitation of one or more claims of the '448 Patent. Such products include at least all iPhones and iPads running iOS 15 or later (e.g., iPhone 15, iPhone Pro, iPad Pro (11-inch, M4, 2024), iPad Pro (13-inch, M4, 2024)), all Macs running macOS Big Sur or later (e.g., iMac (24-inch, 2023), MacBook Pro (14-inch, Nov 2023), MacBook Pro (16-inch, Nov 2023), MacBook Air (13-inch, M3, 2024), and MacBook Air (15-inch, M3, 2024)), and all Apple TVs running tvOS 14 or later, in addition to Apple software and services, such as Safari, QuickTime, and Apple TV+ (the '448 Accused Products) (https://www.coconut.co/articles/ultimate-guide-apples-av1-support-2023; https://bitmovin.com/av1-playback-support), each of which is a moving picture encoding system that makes an encoding of a sequence of moving pictures with a resolution higher than a standard resolution using moving pictures contents which include a sequence of moving pictures

with the standard resolution and do not include a sequence of moving pictures with a resolution higher than the standard resolution, the moving picture encoding system comprising: a first encoder configured to work on a sequence of moving pictures with a standard resolution to implement a first combination of processes for an encoding and a decoding to create a first sequence of encoded bits and a set of decoded pictures with the standard resolution; a first super-resolution enlarger configured to work on the sequence of moving pictures with the standard resolution to implement a process for a first super-resolution enlargement to create a set of super-resolution enlarged pictures with a resolution higher than the standard resolution; a second super-resolution enlarger configured to acquire the set of decoded pictures from the first encoder to implement thereon a process for a second super-resolution enlargement to create a set of super-resolution enlarged decoded pictures with a resolution higher than the standard resolution; a third resolution converter configured to acquire the set of decoded pictures from the first encoder to implement thereon a process for a third resolution conversion to create a set of resolution converted enlarged decoded pictures with a resolution higher than the standard resolution; and a third encoder configured to have the set of super-resolution enlarged pictures from the first super-resolution enlarger as a set of encoding target pictures, employing the set of super-resolution enlarged decoded pictures from the second super-resolution enlarger and the set of resolution converted enlarged decoded pictures from the third resolution converter as sets of reference pictures, to implement thereon a third combination of processes for a prediction and an encoding to create a third sequence of encoded bits, wherein a spatial resolution of the set of super-resolution enlarged pictures, that of the set of super-resolution enlarged decoded pictures, and that of the set of resolution converted enlarged decoded pictures are made equal; and wherein the third encoder controls a selection of a set of reference pictures and creates a set of data on the selection of the set of reference pictures to identify a selected set of reference pictures during the process for prediction of the third combination.

30. For example, Defendant has and continues to directly infringe at least claim 1 of the ’448 Patent by making, using, offering to sell, selling, and/or importing into the United States the ’448 Accused Products.

31. Each of the ’448 Accused Products is a moving picture encoding system that makes an encoding of a sequence of moving pictures with a resolution higher than a standard resolution using moving pictures contents which include a sequence of moving pictures with the standard resolution and do not include a sequence of moving pictures with a resolution higher than the standard resolution, the moving picture encoding system. For example, the libaom-av1 encoder, an implementation of the AV1 encoder specification developed and used by the ’448 Accused Products, allows the input video pictures at a higher resolution to be coded at a low resolution. Libaom-av1 allows encoding with different sizes of reference frames (i.e., upscaled or downscaled frames). The encoding pipeline in libaom-av1 consists of different open-loop and closed-loop processes.

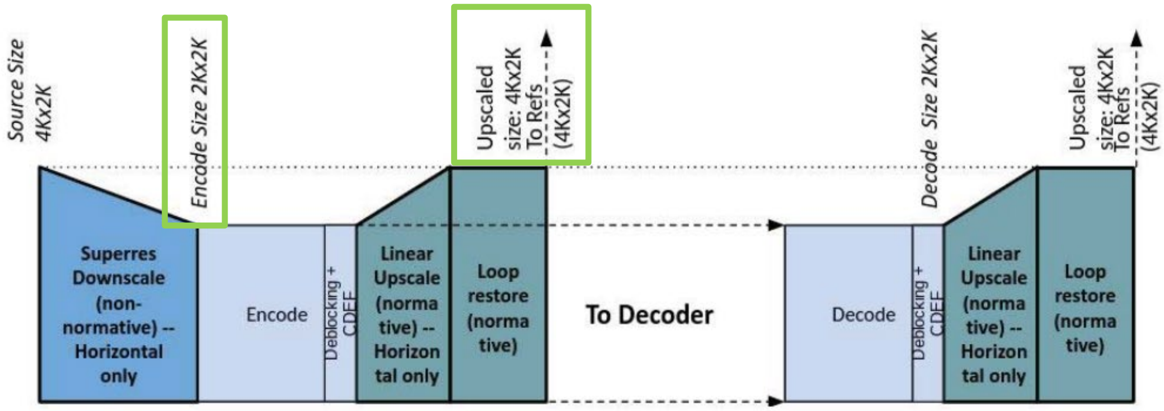
***Abstract*— AV1 is a recently standardized royalty-free video codec from the industry consortium Alliance for Open Media. One of the most innovative coding tools supported in AV1 is an in-loop frame super-resolution mode, that allows an encoder to code any frame at a horizontally reduced spatial resolution by one of several levels, followed by upsampling and super-resolving to full resolution, before replacing reference buffers. This mode is partly enabled by a feature in AV1 that natively allows the motion compensated prediction loop to operate across scales between a coded frame and the available references, thereby allowing on-the-fly resolution change mid-stream within a sequence. For the actual super-resolving process a normative upscaler is followed by an in-loop restoration tool that recovers some of the high frequency information lost in the downsampling process. On-the-fly resolution change capability in conjunction with the frame-superresolution mode in AV1 opens up a new dimension for codec bitrate and quality optimization that has not been possible to explore in any prior standardized video codec with (soon expected) decoding hardware support. This paper provides an overview of how the relevant tools work in AV1, but unlocking them with intelligent encoder decisions to extract real-world benefit, is largely left as future work.**

Source: IEEE, "In-loop Frame Super-resolution in AV1," at 1 (available at https://ieeexplore.ieee.org/document/8954553).

3.7.4 Frame super-resolution

To improve the perceptual quality of decoded pictures, a super-resolution process can be applied at low bitrates. The super-resolution process can be performed on a per-frame basis. First, at the encoder side, the source video is downscaled as a nonnormative procedure. Second, the downscaled video is encoded, followed by deblocking and CDEF filtering processes. Third, a linear upscaling process is applied as a normative procedure to convert the encoded video back to its original resolution. Finally, an LR filter is applied to further recover some lost details of the upscaled picture. The last two steps make up the super-resolving process; the deblocking and CDEF filters are applied at lower spatial resolution on the decoder side, after which the frames are passed through the super-resolution process. To reduce complexity regarding the use of line buffers in hardware implementation, the upscaling and downscaling processes are applied to the horizontal dimension only.

Source: https://aomedia.org/docs/AV1_ToolDescription_v11-clean.pdf, 36.



Source: IEEE, "In-loop Frame Super-resolution in AV1," at 3 (available at https://ieeexplore.ieee.org/document/8954553).

resolution. Therefore, at any point during the encoding and decoding process, any inter frame could be predicted from references that are at different resolutions, and consequently a normative mechanism to predict a block in that frame from a different resolution reference buffer needs to be defined. In principle, as long as we have defined a normative upscaler and a normative downscaler, such prediction across scales would be possible to support. However it would be

Source: IEEE, "In-loop Frame Super-resolution in AV1," at 4 (available at https://ieeexplore.ieee.org/document/8954553).

Supported video codecs (AU7/BEA Series)

File format	Container	Video codecs	Resolution	Frame rate (fps)	Bitrate (Mbps)	Audio codecs
*.avi *.mkv *.asf *.wmv *.mp4 *.mov *.3gp *.vro *.mpg *.mpeg *.ts *.tp *.trp *.flv *.vob *.svi *.m2ts *.mts	AVI MKV ASF MP4 3GP MOV FLV VRO VOB PS TS SVAF	H.264 BP/MP/HP	3840 x 2160	3840 x 2160: 30 1920 x 1080: 60	50	Dolby Digital LPCM ADPCM(IMA, MS) AAC HE-AAC WMA Dolby Digital+ MPEG(MP3) AC-4 G.711(A-Law, μ-Law) OPUS
		HEVC (H.265 - Main, Main10)	3840 x 2160	60	50	
		Motion JPEG	1920 x 1080	60	20	
		MVC	1920 x 1080	60	20	
		MPEG4 SP/ASP	1920 x 1080	60	20	
		Window Media Video v9 (VC1)	1920 x 1080	60	20	
		MPEG2	1920 x 1080	60	20	
		MPEG1	1920 x 1080	60	20	
		Microsoft MPEG-4 v1, v2, v3	1920 x 1080	30	20	
		Window Media Video v7 (WMV1), v8 (WMV2)	1920 x 1080	30	20	
		H.263 Sorrenson	1920 x 1080	30	20	
		VP6	1920 x 1080	30	20	
		AV1	3840 x 2160	60	40	
*.webm	WebM	VP8	1920 x 1080	60	20	Vorbis
		VP9 (Profile 0, profile 2 supported)	3840 x 2160	60	50	
		AV1	3840 x 2160	60	40	
*.rmvb	RMVB	RV8/9/10 (RV30/40)	1920 x 1080	60	20	RealAudio 6

Source: https://downloadcenter.Apple.com/content/UM/202109/20210930143505633/ OSNDVBADA-7.1.0_EM_OSCAR_ASIA_ENG_210709.0.pdf

32. The ’448 Accused Products include a first encoder configured to work on a sequence of moving pictures with a standard resolution to implement a first combination of processes for an encoding and a decoding to create a first sequence of encoded bits and a set of decoded pictures with the standard resolution. The libaom-av1 implementation of AV1 encoder, used by the ’448 Accused Products, includes a first encoder that encodes the images applied at the input. When super-resolution mode is disabled (i.e., in a first pass), the encoder takes the input images at the original resolution and produces encoded images at the same resolution, such as in open loop I-frame encoding:

Supported video codecs (AU7/BEA Series)

File format	Container	Video codecs	Resolution	Frame rate (fps)	Bitrate (Mbps)	Audio codecs
*.avi *.mkv *.asf *.wmv *.mp4 *.mov *.3gp *.vro *.mpg *.mpeg *.ts *.tp *.trp *.flv *.vob *.svi *.m2ts *.mts	AVI MKV ASF MP4 3GP MOV FLV VRO VOB PS TS SVAF	H.264 BP/MP/HP	3840 x 2160	3840 x 2160: 30 1920 x 1080: 60	50	Dolby Digital LPCM ADPCM(IMA, MS) AAC HE-AAC WMA Dolby Digital+ MPEG(MP3) AC-4 G.711(A-Law, μ-Law) OPUS
		HEVC (H.265 - Main, Main10)		60	50	
		Motion JPEG	1920 x 1080	60	20	
		MVC				
		MPEG4 SP/ASP				
		Window Media Video v9 (VC1)				
		MPEG2				
		MPEG1				
		Microsoft MPEG-4 v1, v2, v3		30		
		Window Media Video v7 (WMV1), v8 (WMV2)				
		H.263 Sorrenson				
		VP6				
		AV1	3840 x 2160	60	40	
*.webm	WebM	VP8	1920 x 1080	60	20	Vorbis
		VP9 (Profile 0, profile 2 supported)	3840 x 2160	60	50	
		AV1	3840 x 2160	60	40	
*.rmvb	RMVB	RV8/9/10 (RV30/40)	1920 x 1080	60	20	RealAudio 6

Source: https://downloadcenter.Apple.com/content/UM/202109/20210930143505633/ OSNDVBADA-7.1.0_EM_OSCAR_ASIA_ENG_210709.0.pdf

```
static int denoise_and_encode(AV1_COMP *const cpi, uint8_t *const dest,
                              EncodeFrameInput *const frame_input,
                              EncodeFrameParams *const frame_params,
                              EncodeFrameResults *const frame_results) {
  const AV1EncoderConfig *const oxcf = &cpi->oxcf;
  AV1_COMMON *const cm = &cpi->common;
```

Source: https://github.com/ultrawide/libaom/blob/master/av1/encoder/encodemv.h

```
  if (!encode_show_existing_frame(cm)) {
    // Refresh fb_of_context_type[]: see encoder.h for explanation
    if (cm->current_frame.frame_type == KEY_FRAME) {
      // All ref frames are refreshed, pick one that will live long enough
      fb_of_context_type[current_frame_ref_type] = 0;
    } else {
      // If more than one frame is refreshed, it doesn't matter which one we
      // pick so pick the first.  LST sometimes doesn't refresh any: this is ok

      for (int i = 0; i < REF_FRAMES; i++) {
        if (cm->current_frame.refresh_frame_flags & (1 << i)) {
          fb_of_context_type[current_frame_ref_type] = i;
          break;
        }
      }
    }
  }
}
```

```
void av1_encode_mv(AV1_COMP *cpi, aom_writer *w, const MV *mv, const MV *ref,
                   nmv_context *mvctx, int usehp);
```

Source: https://github.com/ultrawide/libaom/blob/master/av1/encoder/encodemv.h

33. The ’448 Accused Products include a first super-resolution enlarger configured to work on the sequence of moving pictures with the standard resolution to implement a process for a first super-resolution enlargement to create a set of super-resolution enlarged pictures with a resolution higher than the standard resolution. After subsequent processing steps in the AV1 encoding process (e.g., deblocking, CDEF), downscaled or upscaled input images and reference images are upscaled and loop restored for super resolution functionality. Thus, the upscaled and encoded pictures generated based on the downscaled input and reference images are super-resolution enlarged pictures, which have a resolution higher than the input images. The component implementing upscaling of the input images for the above processing step acts as the first resolution enlarger.

A. Scaled Prediction

In order to enable the codec to switch frame resolutions mid-stream, both AV1 and its predecessor VP9 support the ability to predict across scales in the inter prediction loop. As shown schematically in Fig. 1, this allows any frame or frames to be non-normatively downscaled or upscaled (Fig. 1 shows downscaling only) on-the-fly before encoding at a different resolution. The reconstructed frame after encoding at the reduced or increased resolution then replaces one of the reference buffer slots at that resolution. Therefore, at any point during the encoding and decoding process, any inter frame could be predicted from references that are at different resolutions, and consequently a normative mechanism to predict a block in that frame from a different resolution reference buffer needs to be defined. In principle, as long as we have defined a normative upscaler and a normative downscaler, such prediction across scales would be possible to support. However it would be more compute efficient to combine such rescaling with subpel interpolation for motion compensation, and that is what AV1 does.

Source: IEEE, "In-loop Frame Super-resolution in AV1," at 2 (available at https://ieeexplore.ieee.org/document/8954553).

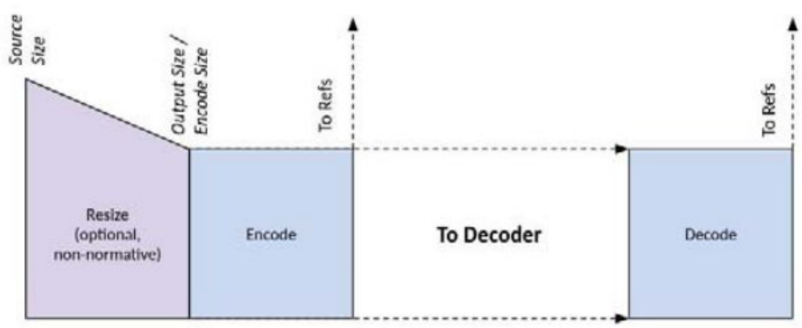


Fig. 1. Scaled prediction framework

Fig: 11, IEEE, "In-loop Frame Super-resolution in AV1," at 2 (available at https://ieeexplore.ieee.org/document/8954553).

```
// Encode frames.
while (aom_img_read(raw, infile) && frame_count < limit) {
  ++frame_count;
  encode_frame(&codec, raw, frame_count, 1, 0, writer);
}
```

Source: https://github.com/ultrawide/libaom/blob/master/examples/twopass_encoder.c

```
void av1_encode_mv(AV1_COMP *cpi, aom_writer *w, const MV *mv, const MV *ref,
                   nmv_context *mvctx, int usehp);
```

Source: https://github.com/ultrawide/libaom/blob/master/av1/encoder/encodemv.h

```
static int encode_frame(aom_codec_ctx_t *ctx, const aom_image_t *img,
                        aom_codec_pts_t pts, unsigned int duration,
                        aom_enc_frame_flags_t flags, AvxVideoWriter *writer) {
  int got_pkts = 0;
  aom_codec_iter_t iter = NULL;
  const aom_codec_cx_pkt_t *pkt = NULL;
  const aom_codec_err_t res = aom_codec_encode(ctx, img, pts, duration, flags);
  if (res != AOM_CODEC_OK) die_codec(ctx, "Failed to encode frame.");

  while ((pkt = aom_codec_get_cx_data(ctx, &iter)) != NULL) {
    got_pkts = 1;
    if (pkt->kind == AOM_CODEC_CX_FRAME_PKT) {
      const int keyframe = (pkt->data.frame.flags & AOM_FRAME_IS_KEY) != 0;

      if (!aom_video_writer_write_frame(writer, pkt->data.frame.buf,
                                        pkt->data.frame.sz,
                                        pkt->data.frame.pts))
        die_codec(ctx, "Failed to write compressed frame.");
      printf(keyframe ? "K" : ".");
      fflush(stdout);
    }
  }
```

Source: https://github.com/ultrawide/libaom/blob/master/examples/twopass_encoder.c

3.7.4 Frame super-resolution

To improve the perceptual quality of decoded pictures, a super-resolution process can be applied at low bitrates. The super-resolution process can be performed on a per-frame basis. First, at the encoder side, the source video is downscaled as a nonnormative procedure. Second, the downscaled video is encoded, followed by deblocking and CDEF filtering processes. Third, a linear upscaling process is applied as a normative procedure to convert the encoded video back to its original resolution. Finally, an LR filter is applied to further recover some lost details of the upscaled picture. The last two steps make up the super-resolving process; the deblocking and

Source: https://aomedia.org/docs/AV1_ToolDescription_v11-clean.pdf, 36

```
// Encode frames.
while (aom_img_read(&raw0, infile0)) {
  int flags = 0;

  // configure and encode base layer

  if (keyframe_interval > 0 && frames_encoded % keyframe_interval == 0)
    flags |= AOM_EFLAG_FORCE_KF;
  else
    // use previous base layer (LAST) as sole reference
    // save this frame as LAST to be used as reference by enhanmcent layer
    // and next base layer
    flags |= AOM_EFLAG_NO_REF_LAST2 | AOM_EFLAG_NO_REF_LAST3 |
             AOM_EFLAG_NO_REF_GF | AOM_EFLAG_NO_REF_ARF |
             AOM_EFLAG_NO_REF_BWD | AOM_EFLAG_NO_REF_ARF2 |
             AOM_EFLAG_NO_UPD_GF | AOM_EFLAG_NO_UPD_ARF |
             AOM_EFLAG_NO_UPD_ENTROPY;
  cfg.g_w = info.frame_width;
  cfg.g_h = info.frame_height;
  if (aom_codec_enc_config_set(&codec, &cfg))
    die_codec(&codec, "Failed to set enc cfg for layer 0");
  if (aom_codec_control(&codec, AOME_SET_SPATIAL_LAYER_ID, 0))
    die_codec(&codec, "Failed to set layer id to 0");
  if (aom_codec_control(&codec, AOME_SET_CQ_LEVEL, 62))
    die_codec(&codec, "Failed to set cq level");
  encode_frame(&codec, &raw0, frame_count++, flags, outfile);
```

Source: https://github.com/ultrawide/libaom/blob/master/examples/twopass_encoder.c

34. The ’448 Accused Products include a second super-resolution enlarger configured to acquire the set of decoded pictures from the first encoder to implement thereon a process for a second super-resolution enlargement to create a set of super-resolution enlarged decoded pictures with a resolution higher than the standard resolution. While in-loop processing for super-resolution functionality is enabled, the libaom-av1 encoder uses the downscaled/upscaled reference images (i.e., reconstructed reference images) and input images. After subsequent processing steps (e.g., deblocking, CDEF) these images are upscaled and loop restored for super resolution. Thus, the upscaled and encoded reference pictures generated based on the downscaled/upscaled input images for encoding of the next set of input images have a resolution higher than the input images. The

component implementing the above processing step acts as the second resolution enlarger.

super-resolution mode in the bit-stream syntax. Specifically, if the super-resolution mode is enabled for a frame, the frame is non-normatively downsampled to a lower resolution and coded at lower resolution using higher resolution references in the motion compensation loop, followed by normative upsampling and super-resolving to generate a full resolution output frame that is displayed and that updates a reference frame store slot for use in subsequent frames. Note also that high-performant CNN based and/or

Source: IEEE, "In-loop Frame Super-resolution in AV1," at 4 (available at https://ieeexplore.ieee.org/document/8954553).

3.7.4 Frame super-resolution

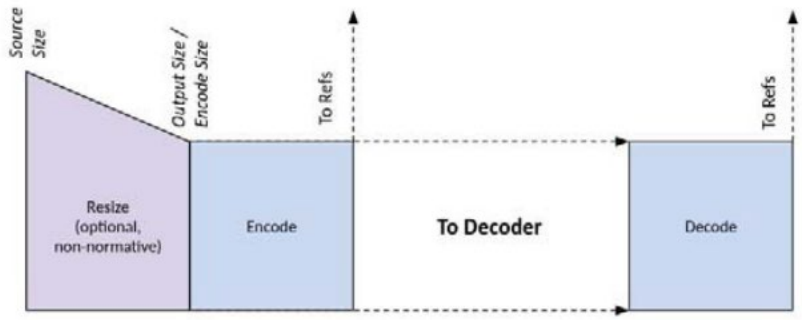
To improve the perceptual quality of decoded pictures, a super-resolution process can be applied at low bitrates. The super-resolution process can be performed on a per-frame basis. First, at the encoder side, the source video is downscaled as a nonnormative procedure. Second, the downscaled video is encoded, followed by deblocking and CDEF filtering processes. Third, a linear upscaling process is applied as a normative procedure to convert the encoded video back to its original resolution. Finally, an LR filter is applied to further recover some lost details of the upscaled picture. The last two steps make up the super-resolving process; the deblocking and

Source: https://aomedia.org/docs/AV1_ToolDescription_v11-clean.pdf, at 36

A. Scaled Prediction

In order to enable the codec to switch frame resolutions mid-stream, both AV1 and its predecessor VP9 support the ability to predict across scales in the inter prediction loop. As shown schematically in Fig. 1, this allows any frame or frames to be non-normatively downscaled or upscaled (Fig. 1 shows downscaling only) on-the-fly before encoding at a different resolution. The reconstructed frame after encoding at the reduced or increased resolution then replaces one of the reference buffer slots at that resolution. Therefore, at any point during the encoding and decoding process, any inter frame could be predicted from references that are at different resolutions, and consequently a normative mechanism to predict a block in that frame from a different resolution reference buffer needs to be defined. In principle, as long as we have defined a normative upscaler and a normative downscaler, such prediction across scales would be possible to support. However it would be more compute efficient to combine such rescaling with subpel interpolation for motion compensation, and that is what AV1 does.

Source: IEEE, "In-loop Frame Super-resolution in AV1," at 2 (available at https://ieeexplore.ieee.org/document/8954553).



Source: IEEE, “In-loop Frame Super-resolution in AV1,” at 2 (available at https://ieeexplore.ieee.org/document/8954553)

```
// Encode frames.
while (aom_img_read(raw, infile) && frame_count < limit) {
  ++frame_count;
  encode_frame(&codec, raw, frame_count, 1, 0, writer);
}
```

Source: https://github.com/ultrawide/libaom/blob/master/examples/twopass_encoder.c

```
void av1_encode_mv(AV1_COMP *cpi, aom_writer *w, const MV *mv, const MV *ref,
                   nmv_context *mvctx, int usehp);
```

Source: https://github.com/ultrawide/libaom/blob/master/av1/encoder/encodemv.h

```c
static int encode_frame(aom_codec_ctx_t *ctx, const aom_image_t *img,
                        aom_codec_pts_t pts, unsigned int duration,
                        aom_enc_frame_flags_t flags, AvxVideoWriter *writer) {
  int got_pkts = 0;
  aom_codec_iter_t iter = NULL;
  const aom_codec_cx_pkt_t *pkt = NULL;
  const aom_codec_err_t res = aom_codec_encode(ctx, img, pts, duration, flags);
  if (res != AOM_CODEC_OK) die_codec(ctx, "Failed to encode frame.");

  while ((pkt = aom_codec_get_cx_data(ctx, &iter)) != NULL) {
    got_pkts = 1;
    if (pkt->kind == AOM_CODEC_CX_FRAME_PKT) {
      const int keyframe = (pkt->data.frame.flags & AOM_FRAME_IS_KEY) != 0;

      if (!aom_video_writer_write_frame(writer, pkt->data.frame.buf,
                                        pkt->data.frame.sz,
                                        pkt->data.frame.pts))
        die_codec(ctx, "Failed to write compressed frame.");
      printf(keyframe ? "K" : ".");
      fflush(stdout);
    }
  }
```

Source: https://github.com/ultrawide/libaom/blob/master/examples/twopass_encoder.c

```
// Encode frames.
while (aom_img_read(&raw, infile)) {
  int flags = 0;
  if (keyframe_interval > 0 && frame_count % keyframe_interval == 0)
    flags |= AOM_EFLAG_FORCE_KF;
  encode_frame(&codec, &raw, frame_count++, flags, writer);
  frames_encoded++;
  if (max_frames > 0 && frames_encoded >= max_frames) break;
}

// Flush encoder.
while (encode_frame(&codec, NULL, -1, 0, writer)) continue;

printf("\n");
fclose(infile);
printf("Processed %d frames.\n", frame_count);

aom_img_free(&raw);
if (aom_codec_destroy(&codec)) die_codec(&codec, "Failed to destroy codec.");

aom_video_writer_close(writer);

return EXIT_SUCCESS;
}
```

Source: https://github.com/ultrawide/libaom/blob/master/examples/simple_encoder.c

35. The '448 Accused Products include a third resolution converter configured to acquire the set of decoded pictures from the first encoder to implement thereon a process for a third resolution conversion to create a set of resolution converted enlarged decoded pictures with a resolution higher than the standard resolution. For example, during the encoding of I-frames, while in loop proceeding for super-resolution functionality is enabled, the libaom-av1 encoder uses the upscaled reference images (i.e., reconstructed reference images) and downscaled input images, since these downscaled input images are at 2K*2K size and encoded using upscaled reconstructed reference images and output images are produced at 4K*2K size, the component implementing the above processing step acts as the 3rd resolution converter. These images are super-resolution converted enlarged pictures, which have a resolution higher than the input images.

```
// Encode frames.
while (aom_img_read(raw, infile) && frame_count < limit) {
  ++frame_count;
  encode_frame(&codec, raw, frame_count, 1, 0, writer);
}
```

Source: https://github.com/ultrawide/libaom/blob/master/examples/twopass_encoder.c

```
void av1_encode_mv(AV1_COMP *cpi, aom_writer *w, const MV *mv, const MV *ref,
                   nmv_context *mvctx, int usehp);
```

Source: https://github.com/ultrawide/libaom/blob/master/av1/encoder/encodemv.h

```
static int encode_frame(aom_codec_ctx_t *ctx, const aom_image_t *img,
                        aom_codec_pts_t pts, unsigned int duration,
                        aom_enc_frame_flags_t flags, AvxVideoWriter *writer) {
  int got_pkts = 0;
  aom_codec_iter_t iter = NULL;
  const aom_codec_cx_pkt_t *pkt = NULL;
  const aom_codec_err_t res = aom_codec_encode(ctx, img, pts, duration, flags);
  if (res != AOM_CODEC_OK) die_codec(ctx, "Failed to encode frame.");

  while ((pkt = aom_codec_get_cx_data(ctx, &iter)) != NULL) {
    got_pkts = 1;
    if (pkt->kind == AOM_CODEC_CX_FRAME_PKT) {
      const int keyframe = (pkt->data.frame.flags & AOM_FRAME_IS_KEY) != 0;

      if (!aom_video_writer_write_frame(writer, pkt->data.frame.buf,
                                        pkt->data.frame.sz,
                                        pkt->data.frame.pts))
        die_codec(ctx, "Failed to write compressed frame.");
      printf(keyframe ? "K" : ".");
      fflush(stdout);
    }
  }
```

Source: https://github.com/ultrawide/libaom/blob/master/examples/twopass_encoder.c

```c
  // Encode frames.
  while (aom_img_read(&raw, infile)) {
    int flags = 0;
    if (keyframe_interval > 0 && frame_count % keyframe_interval == 0)
      flags |= AOM_EFLAG_FORCE_KF;
    encode_frame(&codec, &raw, frame_count++, flags, writer);
    frames_encoded++;
    if (max_frames > 0 && frames_encoded >= max_frames) break;
  }

  // Flush encoder.
  while (encode_frame(&codec, NULL, -1, 0, writer)) continue;

  printf("\n");
  fclose(infile);
  printf("Processed %d frames.\n", frame_count);

  aom_img_free(&raw);
  if (aom_codec_destroy(&codec)) die_codec(&codec, "Failed to destroy codec.");

  aom_video_writer_close(writer);

  return EXIT_SUCCESS;
}
```

Source: https://github.com/ultrawide/libaom/blob/master/examples/simple_encoder.c

3.7.4 Frame super-resolution

To improve the perceptual quality of decoded pictures, a super-resolution process can be applied at low bitrates. The super-resolution process can be performed on a per-frame basis. First, at the encoder side, the source video is downscaled as a nonnormative procedure. Second, the downscaled video is encoded, followed by deblocking and CDEF filtering processes. Third, a linear upscaling process is applied as a normative procedure to convert the encoded video back to its original resolution. Finally, an LR filter is applied to further recover some lost details of the upscaled picture. The last two steps make up the super-resolving process; the deblocking and

Source: https://aomedia.org/docs/AV1_ToolDescription_v11-clean.pdf

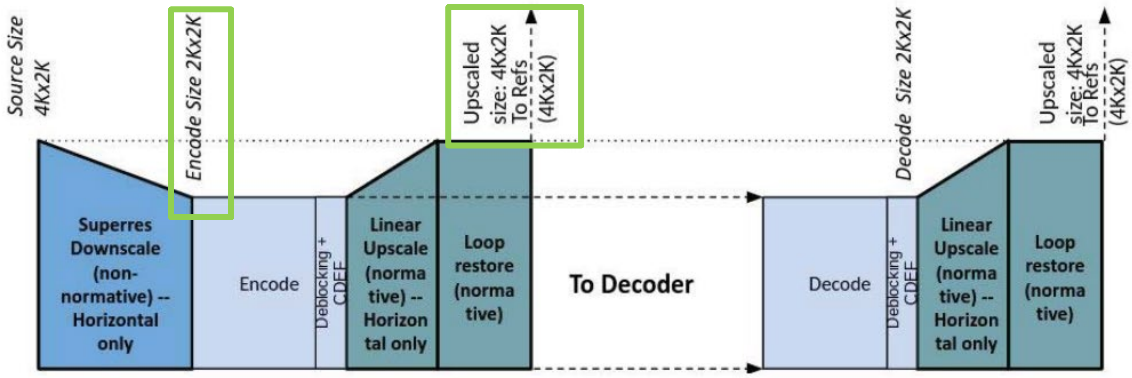


Fig. 4. Frame Super-resolution Framework

Source: IEEE, "In-loop Frame Super-resolution in AV1," at 4 (available at https://ieeexplore.ieee.org/document/8954553)

super-resolution mode in the bit-stream syntax. Specifically, if the super-resolution mode is enabled for a frame, the frame is non-normatively downsampled to a lower resolution and coded at lower resolution using higher resolution references in the motion compensation loop, followed by normative upsampling and super-resolving to generate a full resolution output frame that is displayed and that updates a reference frame store slot for use in subsequent frames. Note also that high-performant CNN based and/or

Source: IEEE, "In-loop Frame Super-resolution in AV1," at 4 (available at https://ieeexplore.ieee.org/document/8954553)

3.7.4 Frame super-resolution

To improve the perceptual quality of decoded pictures, a super-resolution process can be applied at low bitrates. The super-resolution process can be performed on a per-frame basis. First, at the encoder side, the source video is downscaled as a nonnormative procedure. Second, the downscaled video is encoded, followed by deblocking and CDEF filtering processes. Third, a linear upscaling process is applied as a normative procedure to convert the encoded video back to its original resolution. Finally, an LR filter is applied to further recover some lost details of the upscaled picture. The last two steps make up the super-resolving process; the deblocking and

Source: https://aomedia.org/docs/AV1_ToolDescription_v11-clean.pdf, 36

A. Scaled Prediction

In order to enable the codec to switch frame resolutions mid-stream, both AV1 and its predecessor VP9 support the ability to predict across scales in the inter prediction loop. As shown schematically in Fig. 1, this allows any frame or frames to be non-normatively downscaled or upscaled (Fig. 1 shows downscaling only) on-the-fly before encoding at a different resolution. The reconstructed frame after encoding at the reduced or increased resolution then replaces one of the reference buffer slots at that resolution. Therefore, at any point during the encoding and decoding process, any inter frame could be predicted from references that are at different resolutions, and consequently a normative mechanism to predict a block in that frame from a different resolution reference buffer needs to be defined. In principle, as long as we have defined a normative upscaler and a normative downscaler, such prediction across scales would be possible to support. However it would be more compute efficient to combine such rescaling with subpel interpolation for motion compensation, and that is what AV1 does.

Fig: 33, Source: IEEE, "In-loop Frame Super-resolution in AV1," at 2 (available at https://ieeexplore.ieee.org/document/8954553)

36. The '448 Accused Products include a third encoder configured to have the set of super-resolution enlarged pictures from the first super-resolution enlarger as a set of encoding target pictures, employing the set of super-resolution enlarged decoded pictures from the second super-resolution enlarger and the set of resolution converted enlarged decoded pictures from the third resolution converter as sets of reference pictures, to implement thereon a third combination of processes for a prediction and an encoding to create a third sequence of encoded bits. For example, when an I-frame is encoded, the encoder acts as a third encoder, using two reference picture sets (i.e., p-reference-frames and downscaled/upscaled reconstruction reference frames). The p-reference-frames (i.e., first reference picture set that is the output of the first encoder) denotes the list of reference pictures consisting of input pictures at the original resolution. On the other hand, reconstruction reference frames (i.e., second reference set that is the output of second resolution converter) denotes the reference picture list which contains the reconstructed pictures.

```
// Encode frames.
while (aom_img_read(raw, infile) && frame_count < limit) {
  ++frame_count;
  encode_frame(&codec, raw, frame_count, 1, 0, writer);
}
```

Source: https://github.com/ultrawide/libaom/blob/master/examples/twopass_encoder.c

```
void av1_encode_mv(AV1_COMP *cpi, aom_writer *w, const MV *mv, const MV *ref,
                   nmv_context *mvctx, int usehp);
```

Source: https://github.com/ultrawide/libaom/blob/master/av1/encoder/encodemv.h

```
static int encode_frame(aom_codec_ctx_t *ctx, const aom_image_t *img,
                        aom_codec_pts_t pts, unsigned int duration,
                        aom_enc_frame_flags_t flags, AvxVideoWriter *writer) {
  int got_pkts = 0;
  aom_codec_iter_t iter = NULL;
  const aom_codec_cx_pkt_t *pkt = NULL;
  const aom_codec_err_t res = aom_codec_encode(ctx, img, pts, duration, flags);
  if (res != AOM_CODEC_OK) die_codec(ctx, "Failed to encode frame.");

  while ((pkt = aom_codec_get_cx_data(ctx, &iter)) != NULL) {
    got_pkts = 1;
    if (pkt->kind == AOM_CODEC_CX_FRAME_PKT) {
      const int keyframe = (pkt->data.frame.flags & AOM_FRAME_IS_KEY) != 0;

      if (!aom_video_writer_write_frame(writer, pkt->data.frame.buf,
                                        pkt->data.frame.sz,
                                        pkt->data.frame.pts))
        die_codec(ctx, "Failed to write compressed frame.");
      printf(keyframe ? "K" : ".");
      fflush(stdout);
    }
```

Source: https://github.com/ultrawide/libaom/blob/master/examples/twopass_encoder.c

```
  // Encode frames.
  while (aom_img_read(&raw, infile)) {
    int flags = 0;
    if (keyframe_interval > 0 && frame_count % keyframe_interval == 0)
      flags |= AOM_EFLAG_FORCE_KF;
    encode_frame(&codec, &raw, frame_count++, flags, writer);
    frames_encoded++;
    if (max_frames > 0 && frames_encoded >= max_frames) break;
  }

  // Flush encoder.
  while (encode_frame(&codec, NULL, -1, 0, writer)) continue;

  printf("\n");
  fclose(infile);
  printf("Processed %d frames.\n", frame_count);

  aom_img_free(&raw);
  if (aom_codec_destroy(&codec)) die_codec(&codec, "Failed to destroy codec.");

  aom_video_writer_close(writer);

  return EXIT_SUCCESS;
}
```

Source: https://github.com/ultrawide/libaom/blob/master/examples/simple_encoder.c

37. The '448 Accused Products include wherein a spatial resolution of the set of super-resolution enlarged pictures, that of the set of super-resolution enlarged decoded pictures, and that of the set of resolution converted enlarged decoded pictures are made equal. As discussed above, the AV1 encoding process uses downscaled input images which are upscaled using upscaled reconstructed reference images (i.e., reference pictures) and a linear upscale process after deblocking and CDEF steps. Thus, the encoder converts the resolution of applied downscaled input images at various stages to generate the respective upscaled output images (i.e., super-resolution enlarged pictures, super-resolution enlarged decoded pictures, resolution converted enlarged decoded pictures). Since the upscaling is done on the output of all three stages, the resolution of the output images generated via all of the three processes is equal.

3.7.4 Frame super-resolution

To improve the perceptual quality of decoded pictures, a super-resolution process can be applied at low bitrates. The super-resolution process can be performed on a per-frame basis. First, at the encoder side, the source video is downscaled as a nonnormative procedure. Second, the downscaled video is encoded, followed by deblocking and CDEF filtering processes. Third, a linear upscaling process is applied as a normative procedure to convert the encoded video back to its original resolution. Finally, an LR filter is applied to further recover some lost details of the upscaled picture. The last two steps make up the super-resolving process; the deblocking and CDEF filters are applied at lower spatial resolution on the decoder side, after which the frames are passed through the super-resolution process. To reduce complexity regarding the use of line buffers in hardware implementation, the upscaling and downscaling processes are applied to the horizontal dimension only.

Source: https://aomedia.org/docs/AV1_ToolDescription_v11-clean.pdf, 36

38. The '448 Accused Products include a third encoder which controls a selection of a set of reference pictures and creates a set of data on the selection of the set of reference pictures to identify a selected set of reference pictures during the process for prediction of the third combination. As discussed above, the AV1 encoding process uses two sets of reference frames. Since different processes (e.g., motion estimation and mode decision) in encoding use reference pictures from "pa_ref" (i.e., set of first reference pictures) and "recon_ref" (i.e., set of second reference pictures) respectively, it appears that the encoder selects one reference frame from each of the first and second reference picture set, which further constitutes the set of selected reference frames. Further, the reference frame from each first and second set is selected based on the denominator value, which matches the width and current resolution of the input picture.

```
static int choose_primary_ref_frame(
    const AV1_COMP *const cpi, const EncodeFrameParams *const frame_params) {
  const AV1_COMMON *const cm = &cpi->common;

  const int intra_only = frame_params->frame_type == KEY_FRAME ||
                         frame_params->frame_type == INTRA_ONLY_FRAME;
  if (intra_only || frame_params->error_resilient_mode || cpi->use_svc ||
      cpi->ext_flags.use_primary_ref_none) {
    return PRIMARY_REF_NONE;
  }
```

Source: https://github.com/ultrawide/libaom/blob/master/av1/encoder/encode_strategy.c

```
static int get_current_frame_ref_type(
    const AV1_COMP *const cpi, const EncodeFrameParams *const frame_params) {
  // We choose the reference "type" of this frame from the flags which indicate
  // which reference frames will be refreshed by it.  More than one  of these
  // flags may be set, so the order here implies an order of precedence. This is
  // just used to choose the primary_ref_frame (as the most recent reference
  // buffer of the same reference-type as the current frame)
```

Source: https://github.com/ultrawide/libaom/blob/master/av1/encoder/encode_strategy.c

```
void av1_find_best_ref_mvs_from_stack(int allow_hp,
                                      const MB_MODE_INFO_EXT *mbmi_ext,
                                      MV_REFERENCE_FRAME ref_frame,
                                      int_mv *nearest_mv, int_mv *near_mv,
                                      int is_integer);
```

Source: https://github.com/ultrawide/libaom/blob/master/av1/encoder/encode_strategy.c

39. Apple has indirectly infringed and continues to indirectly infringe one or more claims of the '448 Patent, as provided by 35 U.S.C. § 271(b), by inducing infringement by others, such as Apple's customers and end-users, in this Judicial District and elsewhere in the United States. For example, Apple's customers and end-users directly infringe, either literally or under the doctrine of equivalents, through their use of the inventions claimed in the '448 Patent. Apple induces this direct infringement through its affirmative acts of manufacturing, selling, distributing, and/or otherwise making available the '448 Accused Products, and providing instructions, documentation, and other information to customers and end-users suggesting that they use the '448 Accused Products in an infringing manner, including technical support, marketing, product manuals, advertisements, and online documentation. Because of Apple's inducement, Apple's customers and end-users use the '448 Accused Products in a way Apple intends and they directly infringe the '448 Patent. Apple performs these affirmative acts with knowledge of the '448 Patent and with the intent, or willful blindness, that the induced acts directly infringe the '448 Patent.

40. Apple has indirectly infringed and continues to indirectly infringe one or more

claims of the ’448 Patent, as provided by 35 U.S.C. § 271(c), by contributing to direct infringement by others, such as customers and end-users, in this Judicial District and elsewhere in the United States. Apple’s affirmative acts of selling and offering to sell the ’448 Accused Products in this Judicial District and elsewhere in the United States and causing the ’448 Accused Products to be manufactured, used, sold, and offered for sale contribute to others’ use and manufacture of the ’448 Accused Products, such that the ’448 Patent is directly infringed by others. The accused components within the ’448 Accused Products including, but not limited to, software manufactured by Apple, are material to the invention of the ’448 Patent, are not staple articles or commodities of commerce, have no substantial non-infringing uses, and are known by Apple to be especially made or adapted for use in the infringement of the ’448 Patent. Apple performs these affirmative acts with knowledge of the ’448 Patent and with intent, or willful blindness, that they cause the direct infringement of the ’448 Patent.

41. Apple’s infringement of the ’448 Patent is and has been willful. Apple was on notice of the existence of the ’448 Patent and its infringement thereof, or has been willfully blind as to the existence of the ’448 Patent and its infringement thereof. As one example, Defendant is a member of the Alliance for Open Media, the organization that publishes the AV1 Specification. The Alliance for Open Media’s stated goal was to create a video codec that was free of patent licensing obligations associated with prior video codecs. Defendant’s preference would be that its products previously used a video codec called HEVC, and Defendant was motivated to avoid HEVC licensing fees by developing AV1 through the Alliance for Open Media. https://bitmovin.com/apple-joins-av1-codec-consortium. The Alliance for Open Media, including Defendant, conducted a “comprehensive evaluation of the video codec patent landscape and performance of patent due diligence by world-class codec engineers and legal professionals during

the development stage." https://aomedia.org/press%20releases/the-alliance-for-open-media-statement/. Upon information and belief, this "patent due diligence" either uncovered the existence of the '448 Patent and Defendant's infringement thereof, or should have uncovered the existence of the '448 Patent and Defendant's infringement thereof. Defendant could not have reasonably believed that the development of the AV1 video codec could not infringe any valid patent claims, including those of the '448 Patent.

42. Apple has had actual notice of the '448 Patent from either related prior litigations accusing products with similar AV1 functionalities involving direct competitors of Defendant and/or from its knowledge of ACT's patent portfolio in *Advanced Coding Technologies, LLC v. Apple, Inc.*, Case No. 2:24-cv-00572-JRG (E.D. Tex. July 22, 2024).

43. ACT has suffered damages as a result of Defendant's direct, indirect, and willful infringement of the '448 Patent in an amount to be proved at trial.

COUNT II
(Infringement of the '101 Patent)

44. Paragraphs 1 through 26 are incorporated by reference as if fully set forth herein.

45. ACT has not licensed or otherwise authorized Defendant to make, use, offer for sale, sell, or import any products that embody the inventions of the '101 Patent.

46. Defendant has and continues to directly infringe the '101 Patent, either literally or under the doctrine of equivalents, without authority and in violation of 35 U.S.C. § 271, by making, using, offering to sell, selling, and/or importing into the United States products that satisfy each and every limitation of one or more claims of the '101 Patent. Such products include at least Apple's systems and devices (Macs, iPhones, iPads, iPod Touch, AppleTV) that perform network content delivery, including using iCloud, HTTP Live Streaming, and Apple HomeKit Secure Video (the '101 Accused Products), which includes a server device for media, the server device

for media comprising: an internal storage device for storing digital contents, wherein the server device for media responds to a data transmission request from a network player by stream-delivering corresponding data in corresponding digital contents from the internal storage device to the network player during connection to a network; a transfer control unit adapted to transfer and store part of held digital contents in the internal storage device to a network storage device, wherein the network storage device is connected to the network and is capable of storing data, and wherein said transfer control unit does not transfer, from the internal storage device to the network storage device, the digital contents that cannot be recovered if a network failure occurs during the transferring of the digital contents from the internal storage device to the network storage device; a list information transmission unit adapted to respond to a list presentation request for the held digital contents of the server device for media from the network player by transmitting list information to the network player, wherein the list information lists the digital contents left in the internal storage device and the digital contents transferred from the internal storage device to the network storage device and stored in the network storage device, and wherein the list information maintains a tree structure of the digital contents in the internal storage device before transferring the digital contents to the network storage device; a search unit adapted to respond to a data transmission request for the held digital contents from the network player by searching for a location where the held digital contents are currently stored; and a digital contents data transmission processing unit adapted to allow the corresponding data in held digital contents to be stream-delivered from the network storage device to the network player, if the result of search shows the network storage device, wherein the server device for media is a media player.

47. For example, Defendant has and continues to directly infringe at least claim 1 of the ’101 Patent by making, using, offering to sell, selling, and/or importing into the United States

the ’101 Accused Products.

48. The ’101 Accused Products include a server device for media, the server device for media comprising an internal storage device for storing digital contents. For example, an Apple product, such as a MacBook, iPhone, iPad, iPod Touch, or Apple TV, uses cloud storage such as iCloud to store and deliver the content upon request. The content to be streamed is stored at Apple web servers (i.e., a server device for media which is equipped with an internal storage device for storing digital contents) and delivered to the client device. While delivering the content Apple uses content caching technology to copy the accessed content from the server to local network cache storage efficient access of the content.

Types of Cached Content

Content caching includes, but is not limited to, these content types:

macOS

- Apple Books content
- Apps and app updates from the Mac App Store
- GarageBand downloadable content
- iCloud data caching (photos and documents)
- macOS installers downloaded from the App Store or `softwareupdate --fetch-full installer`
- macOS updates and Internet Recovery images
- Xcode downloadable components such as simulators
- Rosetta

iOS, iPadOS, and tvOS

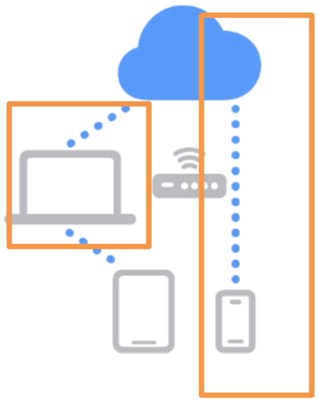
- Apple Books content
- Apple TV screensavers
- Apple TV updates (over the air)
- Certain mobile assets, such as Siri high-quality voices and language dictionaries, and more
- iCloud data caching (photos and documents) for iOS and iPadOS
- iOS and iPadOS updates (over the air)
- iPhone, iPad, and Apple TV apps and app updates
- On-demand resources support for iOS, iPadOS, and tvOS

Source: https://it-training.apple.com/tutorials/deployment/dm070/

> *With content caching*, when a device on your network downloads content from an Apple content server, the content caching service running on your Mac keeps a copy of the content. When another device on your network requests to download the same content, the update is served from the content cache rather than downloaded from the Apple content server.

Source: https://it-training.apple.com/tutorials/deployment/dm070/

49. The ’101 Accused Products include a server device for media responding to a data transmission request from a network player by stream-delivering corresponding data in corresponding digital contents from the internal storage device to the network player during connection to a network. For example, when a client device (i.e., network player) requests content from an Apple content server storage (i.e., internal storage), the Apple server responds by delivering the content to the client device (i.e., network player) requesting the content (i.e., responds to a data transmission request from a network player by stream-delivering corresponding data in corresponding digital contents from the internal storage device to the network player during connection to a network).



With content caching, when a device on your network downloads content from an Apple content server, the content caching service running on your Mac keeps a copy of the content. When another device on your network requests to download the same content, the update is served from the content cache rather than downloaded from the Apple content server.

Source: https://it-training.apple.com/tutorials/deployment/dm070/

When an Apple device on your network tries to download Apple content that could be cached, the Apple content server instructs the device to check with the local network's cache first. If the content isn't available there, the content cache requests the content from the Apple server and stores it in the local network's cache. That content is then available for other Apple devices to retrieve without downloading it from the internet. Because a local network normally shares data much faster than the internet, subsequent devices can download cached content faster. Supported content includes operating-system updates, apps, books, iCloud content, and more.

Source: https://it-training.apple.com/tutorials/deployment/dm070/

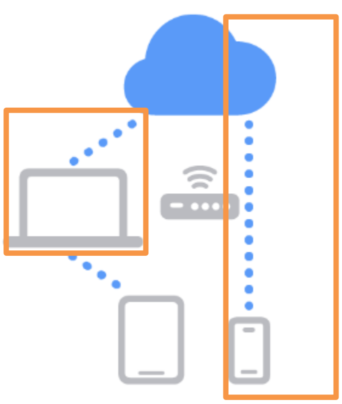
50. The '101 Accused Products include a transfer control unit adapted to transfer and store part of held digital contents in the internal storage device to a network storage device, wherein the network storage device is connected to the network and is capable of storing data, and wherein said transfer control unit does not transfer, from the internal storage device to the network storage device, the digital contents that cannot be recovered if a network failure occurs during the transferring of the digital contents from the internal storage device to the network storage device. For example, when a client device sends a request to access the content file (i.e., digital content) stored at the Apple server (i.e.server device), the response delivers the content file and saves/caches a copy of the content to the local network storage. For ex., when a user accesses the content stored at the Apple server and the requested content is not present at the caching

infrastructure, the requested content is delivered from the origin servers and cached at caching infrastructure such as local network storage (i.e., transferring and storing part of held digital contents in the internal storage device to a network storage device). In addition to this, due to certain policy restrictions, not all the content is allowed to be cached (i.e., and wherein the digital contents that cannot be recovered if a network failure occurs during the transferring of the digital contents are not transferred from the internal storage device to the network storage device;) therefore the server device avoids transferring that non-cachable content to caching infrastructure including CDN (i.e., the digital contents are not transferred from the internal storage device to the network storage device).

How content caching works

After you turn on content caching on a Mac, the Mac keeps a copy of all content that devices (called *clients*) on the local network can download. Content can also be downloaded from multiple iPhone or iPad devices if they're tethered to a Mac using a cart or USB hub. You can specify ranges of client IP addresses (for example, one or two subnets) that a content cache is best positioned to serve, and optionally you can make that content exclusive to those clients by choosing the "Devices using custom local networks" option. The options are:

Source: https://support.apple.com/en-in/guide/deployment/depde72e125f/web



With content caching, when a device on your network downloads content from an Apple content server, the content caching service running on your Mac keeps a copy of the content. When another device on your network requests to download the same content, the update is served from the content cache rather than downloaded from the Apple content server.

Source: https://it-training.apple.com/tutorials/deployment/dm070/

When an Apple device on your network tries to download Apple content that could be cached, the Apple content server instructs the device to check with the local network's cache first. If the content isn't available there, the content cache requests the content from the Apple server and stores it in the local network's cache. That content is then available for other Apple devices to retrieve without downloading it from the internet. Because a local network normally shares data much faster than the internet, subsequent devices can download cached content faster. Supported content includes operating-system updates, apps, books, iCloud content, and more.

Source: https://it-training.apple.com/tutorials/deployment/dm070/

For example, when the first client on your network downloads a macOS update, the content cache keeps a copy of the update. When the next client on the network connects to the App Store to download the update, the update is copied from the content cache rather than from the App Store.

Because the local network is normally much faster than the internet, the second client (and all subsequent clients) download updates much faster.

Source: https://support.apple.com/en-in/guide/deployment/depde72e125f/web

Country and region restrictions

- Not all content is cached in all regions.
- iTunes downloads are not cached in Brazil, Mexico, China mainland, or Portugal. Apple Books downloads are not cached in Canada.
- When macOS content caching is enabled, the IP address and region of a Mac computer are registered with Apple. To cache iTunes and App Store content, the registered region of the macOS content cache must match the region of the client Apple ID accounts.

Source: https://support.apple.com/en-us/102860

51. The '101 Accused Products include a list information transmission unit adapted to respond to a list presentation request for the held digital contents of the server device for media from the network player by transmitting list information to the network player, wherein the list information lists the digital contents left in the internal storage device and the digital contents transferred from the internal storage device to the network storage device and stored in the network storage device, and wherein the list information maintains a tree structure of the digital contents

in the internal storage device before transferring the digital contents to the network storage device. For example, when a client device (e.g., an iPhone) requests to present the list of photos via its photos application (i.e., Network Player) the Apple server (e.g., iCloud) presents the application with a list of photos if some of these photos are already accessed by the device and if the device uses the content cache the list also includes the content stored in the local storage as well server. Additionally, the list of content stored at local network storage can also be accessed. Thus, the device lists both the content stored at the Apple servers as well as the content present at the local network storage. Users may also view the various log matrices that indicate how much content is served from Apple servers, and how much is from local network storage. The properties of such databases indicate that the content list presented is in a structured format and follows the tree structure.

Types of Cached Content

Content caching includes, but is not limited to, these content types:

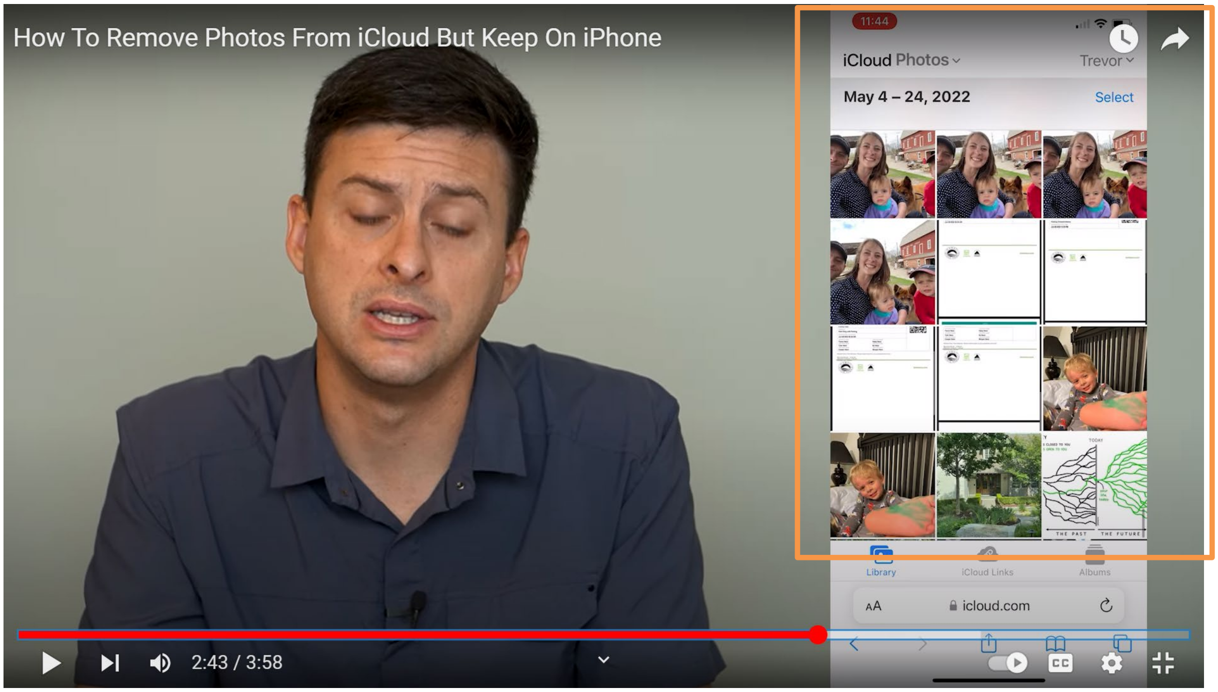
macOS

- Apple Books content
- Apps and app updates from the Mac App Store
- GarageBand downloadable content
- iCloud data caching (photos and documents)
- macOS installers downloaded from the App Store or `softwareupdate --fetch-full installer`
- macOS updates and Internet Recovery images
- Xcode downloadable components such as simulators
- Rosetta

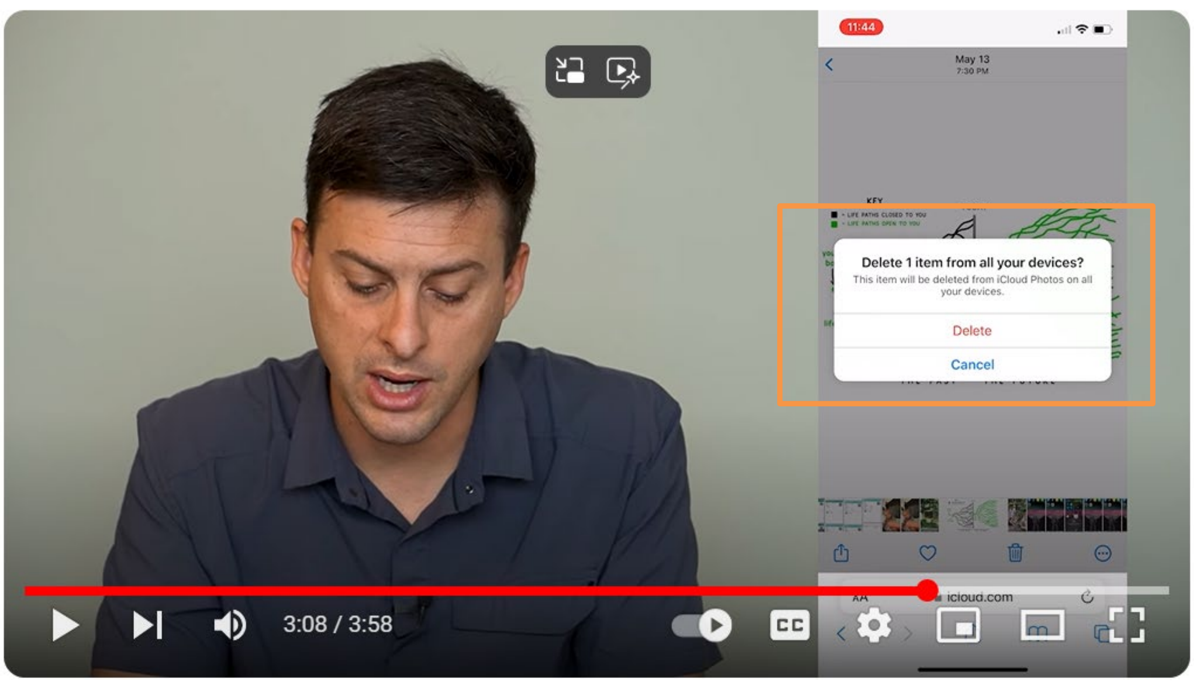
iOS, iPadOS, and tvOS

- Apple Books content
- Apple TV screensavers
- Apple TV updates (over the air)
- Certain mobile assets, such as Siri high-quality voices and language dictionaries, and more
- iCloud data caching (photos and documents) for iOS and iPadOS
- iOS and iPadOS updates (over the air)
- iPhone, iPad, and Apple TV apps and app updates
- On-demand resources support for iOS, iPadOS, and tvOS

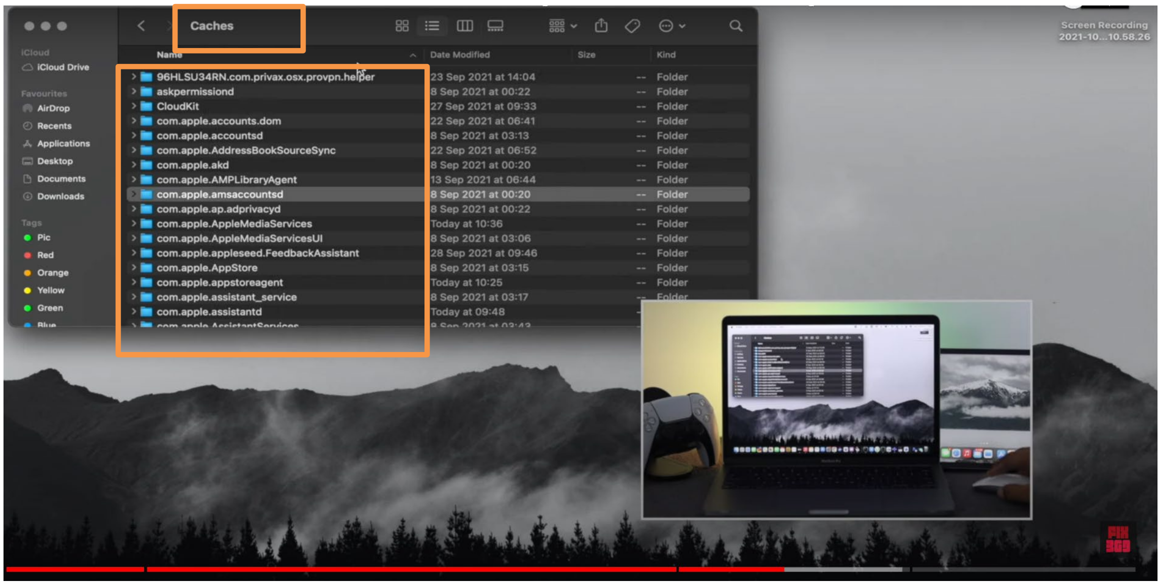
Source: https://it-training.apple.com/tutorials/deployment/dm070/



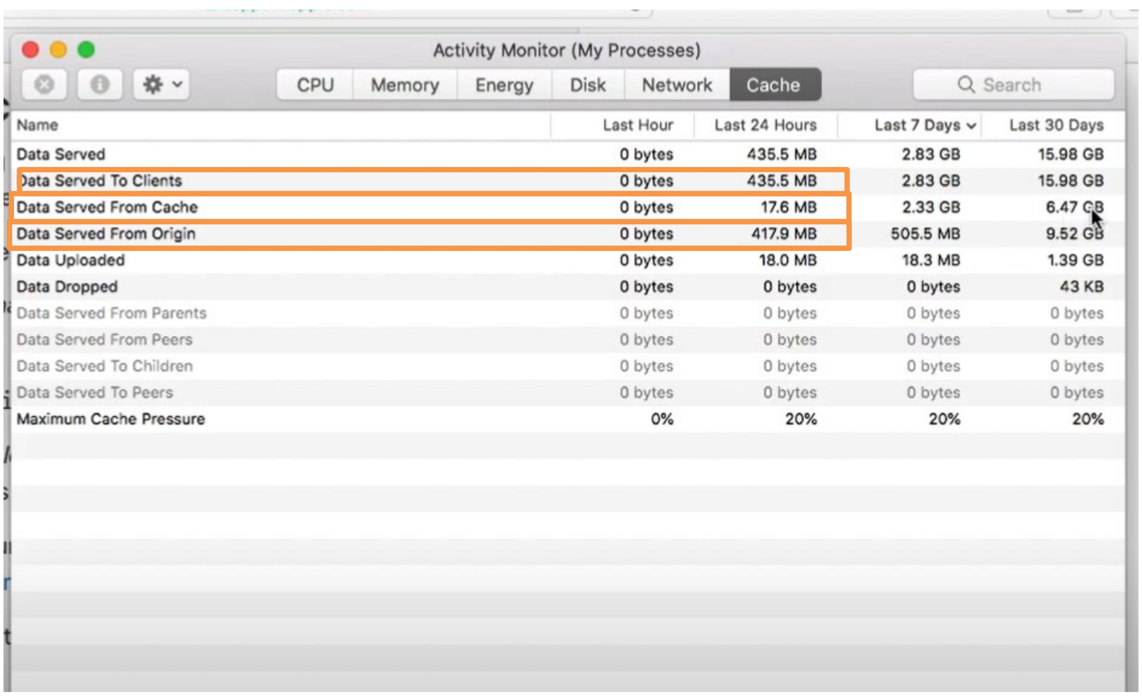
Source: https://www.youtube.com/watch?v=JTpja1TGIWc, at 2:43



Source: https://www.youtube.com/watch?v=JTpja1TGIWc, at 3:08



Source: https://www.youtube.com/watch?v=K1yX1n360dU, at 2:37



Source: https://www.youtube.com/watch?v=babMxI-eh3E, at 4:53

52. The ’101 Accused Products include a search unit adapted to respond to a data transmission request for the held digital contents from the network player by searching for a location where the held digital contents are currently stored. For example, when an Apple device on a network tries to download Apple content that could be cached, the Apple content server

instructs the device to search the content with the local network's cache first. If the content is not available there, the content cache requests the content from the Apple server and stores it in the local network's cache.

When an Apple device on your network tries to download Apple content that could be cached, the Apple content server instructs the device to check with the local network's cache first. If the content isn't available there, the content cache requests the content from the Apple server and stores it in the local network's cache. That content is then available for other Apple devices to retrieve without downloading it from the internet. Because a local network normally shares data much faster than the internet, subsequent devices can download cached content faster. Supported content includes operating-system updates, apps, books, iCloud content, and more.

Source: https://it-training.apple.com/tutorials/deployment/dm070/

53. The '101 Accused Products include a digital contents data transmission processing unit adapted to allow the corresponding data in held digital contents to be stream-delivered from the network storage device to the network player, if the result of search shows the network storage device. For example, when an Apple device on your network tries to download Apple content that could be cached, the Apple content server instructs the device to check with the local network's cache first. If the content is available with the content cache it is delivered to the client device from the local network's cache (i.e., allowing the corresponding data in held digital contents to be stream-delivered from the network storage device to the network player).

How content caching works

After you turn on content caching on a Mac, the Mac keeps a copy of all content that devices (called *clients*) on the local network can download. Content can also be downloaded from multiple iPhone or iPad devices if they're tethered to a Mac using a cart or USB hub. You can specify ranges of client IP addresses (for example, one or two subnets) that a content cache is best positioned to serve, and optionally you can make that content exclusive to those clients by choosing the "Devices using custom local networks" option. The options are:

Source: https://support.apple.com/en-in/guide/deployment/depde72e125f/web

With content caching, when a device on your network downloads content from an Apple content server, the content caching service running on your Mac keeps a copy of the content. When another device on your network requests to download the same content, the update is served from the content cache rather than downloaded from the Apple content server.

Source: https://it-training.apple.com/tutorials/deployment/dm070/

You can use content caching on networks that use network address translation (NAT) for the content cache and all devices, on networks consisting of publicly routable IP addresses, and optionally for devices tethered to a Mac (for example, when provisioning many devices at once using Apple Configurator). Apple devices automatically contact a nearby content cache without any configuration by using a lookup service that maps client private and public IP addresses to configurations registered with Apple from Mac computers with content caching turned on. Because any Apple device on a network silently and automatically uses content caching if available, detailed information on individual assets requested by individual specific clients is not available for privacy reasons. You can, however, query aggregate content caching usage statistics to help measure and gauge performance. For more information, see Plan for and set up content caching.

Source: https://support.apple.com/en-in/guide/deployment/depde72e125f/web

When an Apple device on your network tries to download Apple content that could be cached, the Apple content server instructs the device to check with the local network's cache first. If the content isn't available there, the content cache requests the content from the Apple server and stores it in the local network's cache. That content is then available for other Apple devices to retrieve without downloading it from the internet. Because a local network normally shares data much faster than the internet, subsequent devices can download cached content faster. Supported content includes operating-system updates, apps, books, iCloud content, and more.

Source: https://it-training.apple.com/tutorials/deployment/dm070/

For example, when the first client on your network downloads a macOS update, the content cache keeps a copy of the update. When the next client on the network connects to the App Store to download the update, the update is copied from the content cache rather than from the App Store.

Because the local network is normally much faster than the internet, the second client (and all subsequent clients) download updates much faster.

Source: https://support.apple.com/en-in/guide/deployment/depde72e125f/web

54. The ’101 Accused Products include a server device for media, which is a media player. For example, the content at Apple’s servers can be played/streamed using various client software (e.g., iTunes, AppleTV, etc.) developed by Apple. The Apple servers (i.e., server devices for media) in integration with such client software, play and/or stream the media files. Therefore, Apple servers act as a media player.

Find and watch iTunes movies and TV shows

1. Open iTunes Movies or TV Shows on Apple TV.
2. In the menu bar, navigate to any of the following categories:

   - *Purchased:* See the movies you’ve purchased on the iTunes Store, including purchases made on other iOS or iPadOS devices and purchases by Family Sharing members.
   - *Top Movies* or *Top TV Shows:* Browse the top items in the iTunes Store.
   - *Wish List/Favourites:* Find items you’ve added to your Wish List (Movies) or Favourites (TV shows) but haven’t yet purchased or rented.

Source: https://support.apple.com/en-in/guide/tv/atvb53d6cfb4/tvos

Types of Cached Content

Content caching includes, but is not limited to, these content types:

macOS

- Apple Books content
- Apps and app updates from the Mac App Store
- GarageBand downloadable content
- iCloud data caching (photos and documents)
- macOS installers downloaded from the App Store or `softwareupdate --fetch-full installer`
- macOS updates and Internet Recovery images
- Xcode downloadable components such as simulators
- Rosetta

iOS, iPadOS, and tvOS

- Apple Books content
- Apple TV screensavers
- Apple TV updates (over the air)
- Certain mobile assets, such as Siri high-quality voices and language dictionaries, and more
- iCloud data caching (photos and documents) for iOS and iPadOS
- iOS and iPadOS updates (over the air)
- iPhone, iPad, and Apple TV apps and app updates
- On-demand resources support for iOS, iPadOS, and tvOS

Source: https://it-training.apple.com/tutorials/deployment/dm070/

With content caching, when a device on your network downloads content from an Apple content server, the content caching service running on your Mac keeps a copy of the content. When another device on your network requests to download the same content, the update is served from the content cache rather than downloaded from the Apple content server.

Source: https://it-training.apple.com/tutorials/deployment/dm070/.

55. Apple has indirectly infringed and continues to indirectly infringe one or more claims of the ’101 Patent, as provided by 35 U.S.C. § 271(b), by inducing infringement by others, such as Apple’s customers and end-users, in this Judicial District and elsewhere in the United States. For example, Apple’s customers and end-users directly infringe, either literally or under the doctrine of equivalents, through their use of the inventions claimed in the ’101 Patent. Apple

induces this direct infringement through its affirmative acts of manufacturing, selling, distributing, and/or otherwise making available the '101 Accused Products, and providing instructions, documentation, and other information to customers and end-users suggesting that they use the '101 Accused Products in an infringing manner, including technical support, marketing, product manuals, advertisements, and online documentation. Because of Apple's inducement, Apple's customers and end-users use the '101 Accused Products in a way Apple intends and they directly infringe the '101 Patent. Apple performs these affirmative acts with knowledge of the '101 Patent and with the intent, or willful blindness, that the induced acts directly infringe the '101 Patent.

56. Apple has indirectly infringed and continues to indirectly infringe one or more claims of the '101 Patent, as provided by 35 U.S.C. § 271(c), by contributing to direct infringement by others, such as customers and end-users, in this Judicial District and elsewhere in the United States. Apple's affirmative acts of selling and offering to sell the '101 Accused Products in this Judicial District and elsewhere in the United States and causing the '101 Accused Products to be manufactured, used, sold, and offered for sale contribute to others' use and manufacture of the '101 Accused Products, such that the '101 Patent is directly infringed by others. The accused components within the '101 Accused Products including, but not limited to, software manufactured by Apple, are material to the invention of the '101 Patent, are not staple articles or commodities of commerce, have no substantial non-infringing uses, and are known by Apple to be especially made or adapted for use in the infringement of the '101 Patent. Apple performs these affirmative acts with knowledge of the '101 Patent and with intent, or willful blindness, that they cause the direct infringement of the '101 Patent.

57. Apple has had actual notice of the '101 Patent from either related prior litigations accusing products with similar functionalities involving direct competitors of Defendant and/or

from its knowledge of ACT's patent portfolio in *Advanced Coding Technologies, LLC v. Apple, Inc.*, Case No. 2:24-cv-00572-JRG (E.D. Tex. July 22, 2024).

58. ACT has suffered damages as a result of Defendant's direct, indirect, and willful infringement of the '101 Patent in an amount to be proved at trial.

COUNT III
(Infringement of the '891 Patent)

59. Paragraphs 1 through 26 are incorporated by reference as if fully set forth herein.

60. ACT has not licensed or otherwise authorized Defendant to make, use, offer for sale, sell, or import any products that embody the inventions of the '891 Patent.

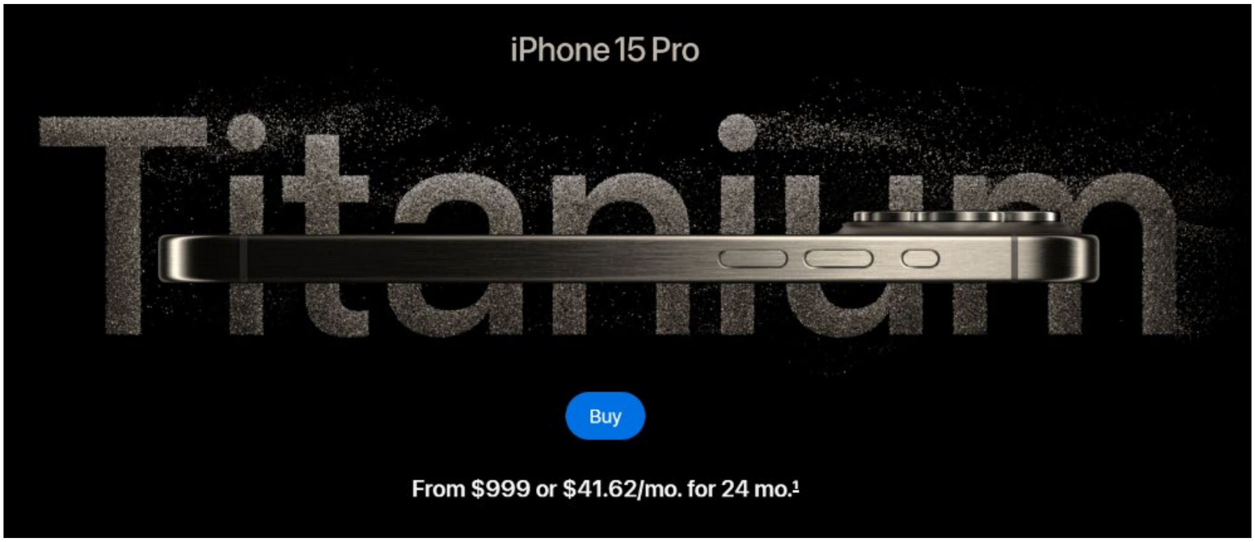
61. Defendant has and continues to directly infringe the '891 Patent, either literally or under the doctrine of equivalents, without authority and in violation of 35 U.S.C. § 271, by making, using, offering to sell, selling, and/or importing into the United States products that satisfy each and every limitation of one or more claims of the '891 Patent. Such products include at least Apple iPhone, iPad, and Apple Watch products compliant with 5G NR, including but not limited to, the iPhone 15, iPhone 15 Plus, iPhone 15 Pro, iPhone 15 Pro Max, iPhone 14, iPhone 14 Plus, iPhone 14 Pro, iPhone 14 Pro Max, iPhone SE (2022), iPhone 13, iPhone 13 Mini, iPhone 13 Pro, iPhone 13 Pro Max, iPhone 12, iPhone 12 Mini, iPhone 12 Pro, iPhone 12 Pro Max, iPad Pro 12.9-inch (5th generation or later), iPad Pro 11-inch (3rd generation or later), iPad Air (5th generation), iPad mini (6th generation), iPad (10th generation), and Apple Watch Series 9 (the '891 Accused Products) which include a communication quality judging device comprising: a symbol judging means for obtaining a baseband signal representative of a sequence of multilevel symbols and judging the symbol represented by the baseband signal; a communication quality judging means for judging communication quality of a transmission channel over which the baseband signal has been transmitted, based on content of the symbol judged by the symbol judging means; and a data

changing means for, if the communication quality judged by the communication quality judging means does not satisfy a predetermined condition, making a predetermined change to the data to be transmitted represented by the symbol used in the judgment, wherein at least a portion of a bit string is distinguished as a protected portion, the bit string constituting data to be transmitted represented by the sequence of symbols, and at least a portion of the symbol that belongs to the sequence of symbols contains a bit belonging to the protected portion and a redundant bit having a predetermined value, and wherein the communication quality judging means identifies the number of redundant bits having the predetermined value or the number of redundant bits missing the predetermined value among the redundant bits contained in the symbol that contains a bit belonging to the protected portion, and judges the communication quality of the transmission channel based on the identified result.

62. For example, Defendant has and continues to directly infringe at least claim 1 of the ’891 Patent by making, using, offering to sell, selling, and/or importing into the United States the ’891 Accused Products.

63. The ’891 Accused Products are each a communication quality judging device comprising a symbol judging means for obtaining a baseband signal representative of a sequence of multilevel symbols and judging the symbol represented by the baseband signal. For example, the accused Apple Pixel 8 Pro supports 5G-NR network connectivity. According to 3GPP TS 38.212, the Low-density Parity-check (LPDC) based Cyclic Redundancy Check (CRC) (i.e., a communication quality judging method) is used to detect any error in the transmitted data. CRC includes obtaining a data and control stream (i.e., a baseband signal) that consists of a plurality of bit streams (i.e., a sequence of multilevel symbols) represented as a_0, a_1…., a_{A-1} and p_0, p_1…, p_{L-1}. The CRC method further includes judging the data and control streams based on these bits (i.e.,

judging the symbols represented by the baseband signal).



Source: https://apple.com/iphone-15-pro/

Cellular and Wireless	**Model A2848*** **Model A2849***	5G NR (Bands n1, n2, n3, n5, n7, n8, n12, n14, n20, n25, n26, n28, n29, n30, n38, n40, n41, n48, n53, n66, n70, n71, n75, n76, n77, n78, n79)
		5G NR mmWave (Bands n258, n260, n261)
		FDD-LTE (Bands 1, 2, 3, 4, 5, 7, 8, 12, 13, 14, 17, 18, 19, 20, 25, 26, 28, 29, 30, 32, 66, 71)
		TD-LTE (Bands 34, 38, 39, 40, 41, 42, 46, 48, 53)
		UMTS/HSPA+/DC-HSDPA (850, 900, 1700/2100, 1900, 2100 MHz)
		GSM/EDGE (850, 900, 1800, 1900 MHz)

Source: https://www.apple.com/iphone-15-pro/specs/

Data and control streams from/to MAC layer are encoded /decoded to offer transport and control services over the radio transmission link. Channel coding scheme is a combination of error detection, error correcting, rate matching, interleaving and transport channel or control information mapping onto/splitting from physical channels.

Source:
https://www.etsi.org/deliver/etsi_ts/138200_138299/138212/15.03.00_60/ts_138212v150300p.pdf, 9.

CRC Cyclic redundancy check

Source:
https://www.etsi.org/deliver/etsi_ts/138200_138299/138212/15.03.00_60/ts_138212v150300p.pdf, 8.

7.2 Downlink shared channel and paging channel

7.2.1 Transport block CRC attachment

Error detection is provided on each transport block through a Cyclic Redundancy Check (CRC).

The entire transport block is used to calculate the CRC parity bits. Denote the bits in a transport block delivered to layer 1 by $a_0, a_1, a_2, a_3, ..., a_{A-1}$, and the parity bits by $p_0, p_1, p_2, p_3, ..., p_{L-1}$, where A is the payload size and L is the number of parity bits. The lowest order information bit a_0 is mapped to the most significant bit of the transport block as defined in Clause 6.1.1 of [TS38.321].

The parity bits are computed and attached to the DL-SCH transport block according to Clause 5.1, by setting L to 24 bits and using the generator polynomial $g_{\text{CRC24A}}(D)$ if $A > 3824$; and by setting L to 16 bits and using the generator polynomial $g_{\text{CRC16}}(D)$ otherwise.

The bits after CRC attachment are denoted by $b_0, b_1, b_2, b_3, ..., b_{B-1}$, where $B = A + L$.

Source:
https://www.etsi.org/deliver/etsi_ts/138200_138299/138212/15.03.00_60/ts_138212v150300p.pdf, 70.

7.2.2 LDPC base graph selection

For initial transmission of a transport block with coding rate R indicated by the MCS index according to Clause 5.1.3.1 in [6, TS 38.214] and subsequent re-transmission of the same transport block, each code block of the transport block is encoded with either LDPC base graph 1 or 2 according to the following:

- if $A \leq 292$, or if $A \leq 3824$ and $R \leq 0.67$, or if $R \leq 0.25$, LDPC base graph 2 is used;
- otherwise, LDPC base graph 1 is used,

Source:
https://www.etsi.org/deliver/etsi_ts/138200_138299/138212/15.03.00_60/ts_138212v150300p.pdf, 71.

5 General procedures

Data and control streams from/to MAC layer are encoded /decoded to offer transport and control services over the radio transmission link. Channel coding scheme is a combination of error detection, error correcting, rate matching, interleaving and transport channel or control information mapping onto/splitting from physical channels.

5.1 CRC calculation

Denote the input bits to the CRC computation by $a_0, a_1, a_2, a_3, ..., a_{A-1}$, and the parity bits by $p_0, p_1, p_2, p_3, ..., p_{L-1}$, where A is the size of the input sequence and L is the number of parity bits. The parity bits are generated by one of the following cyclic generator polynomials:

- $g_{CRC24A}(D) = [D^{24} + D^{23} + D^{18} + D^{17} + D^{14} + D^{11} + D^{10} + D^7 + D^6 + D^5 + D^4 + D^3 + D + 1]$ for a CRC length $L = 24$;
- $g_{CRC24B}(D) = [D^{24} + D^{23} + D^6 + D^5 + D + 1]$ for a CRC length $L = 24$;
- $g_{CRC24C}(D) = [D^{24} + D^{23} + D^{21} + D^{20} + D^{17} + D^{15} + D^{13} + D^{12} + D^8 + D^4 + D^2 + D + 1]$ for a CRC length $L = 24$;
- $g_{CRC16}(D) = [D^{16} + D^{12} + D^5 + 1]$ for a CRC length $L = 16$;
- $g_{CRC11}(D) = [D^{11} + D^{10} + D^9 + D^5 + 1]$ for a CRC length $L = 11$;
- $g_{CRC6}(D) = [D^6 + D^5 + 1]$ for a CRC length $L = 6$.

The encoding is performed in a systematic form, which means that in GF(2), the polynomial:

$$a_0 D^{A+L-1} + a_1 D^{A+L-2} + ... + a_{A-1} D^L + p_0 D^{L-1} + p_1 D^{L-2} + ... + p_{L-2} D^1 + p_{L-1}$$

Source: https://www.etsi.org/deliver/etsi_ts/138200_138299/138212/15.03.00_60/ts_138212v150300p.pdf, 9.

64. The '891 Accused Products include a communication quality judging means for judging communication quality of a transmission channel over which the baseband signal has been transmitted, based on content of the symbol judged by the symbol judging means. According to 3GPP TS 38.212, the Cyclic Redundancy Check (CRC) checks for any error (i.e., judging communication quality) present in the data and control stream comprising of a plurality of bits transmitted over a radio transmission link. The error detection comprises judging the bit stream based on the payload size and parity bits, and accordingly selecting the coding rate.

5 General procedures

Data and control streams from/to MAC layer are encoded /decoded to offer transport and control services over the radio transmission link. Channel coding scheme is a combination of error detection, error correcting, rate matching, interleaving and transport channel or control information mapping onto/splitting from physical channels.

5.1 CRC calculation

Denote the input bits to the CRC computation by $a_0, a_1, a_2, a_3, ..., a_{A-1}$, and the parity bits by $p_0, p_1, p_2, p_3, ..., p_{L-1}$, where A is the size of the input sequence and L is the number of parity bits. The parity bits are generated by one of the following cyclic generator polynomials:

- $g_{CRC24A}(D) = [D^{24} + D^{23} + D^{18} + D^{17} + D^{14} + D^{11} + D^{10} + D^7 + D^6 + D^5 + D^4 + D^3 + D + 1]$ for a CRC length $L = 24$;
- $g_{CRC24B}(D) = [D^{24} + D^{23} + D^6 + D^5 + D + 1]$ for a CRC length $L = 24$;
- $g_{CRC24C}(D) = [D^{24} + D^{23} + D^{21} + D^{20} + D^{17} + D^{15} + D^{13} + D^{12} + D^8 + D^4 + D^2 + D + 1]$ for a CRC length $L = 24$;
- $g_{CRC16}(D) = [D^{16} + D^{12} + D^5 + 1]$ for a CRC length $L = 16$;
- $g_{CRC11}(D) = [D^{11} + D^{10} + D^9 + D^5 + 1]$ for a CRC length $L = 11$;
- $g_{CRC6}(D) = [D^6 + D^5 + 1]$ for a CRC length $L = 6$.

The encoding is performed in a systematic form, which means that in GF(2), the polynomial:

$$a_0 D^{A+L-1} + a_1 D^{A+L-2} + ... + a_{A-1} D^L + p_0 D^{L-1} + p_1 D^{L-2} + ... + p_{L-2} D^1 + p_{L-1}$$

Source: https://www.etsi.org/deliver/etsi_ts/138200_138299/138212/15.03.00_60/ts_138212v150300p.pdf, 9.

7.1.3 Transport block CRC attachment

Error detection is provided on BCH transport blocks through a Cyclic Redundancy Check (CRC).

The entire transport block is used to calculate the CRC parity bits. The input bit sequence is denoted by $a'_0, a'_1, a'_2, a'_3, ..., a'_{A-1}$, and the parity bits by $p_0, p_1, p_2, p_3, ..., p_{L-1}$, where A is the payload size and L is the number of parity bits.

The parity bits are computed and attached to the BCH transport block according to Subclause 5.1 by setting L to 24 bits and using the generator polynomial $g_{CRC24C}(D)$, resulting in the sequence $b_0, b_1, b_2, b_3, ..., b_{B-1}$, where $B = A + L$.

The bit sequence $b_0, b_1, b_2, b_3, ..., b_{B-1}$ is the input bit sequence $c_0, c_1, c_2, c_3, ..., c_{K-1}$ to the channel encoder, where $c_i = b_i$ for $i = 0, 1, ..., B-1$ and $K = B$.

Source: https://www.etsi.org/deliver/etsi_ts/138200_138299/138212/15.03.00_60/ts_138212v150300p.pdf, 70.

7.2.2 LDPC base graph selection

For initial transmission of a transport block with coding rate R indicated by the MCS index according to Clause 5.1.3.1 in [6, TS 38.214] and subsequent re-transmission of the same transport block, each code block of the transport block is encoded with either LDPC base graph 1 or 2 according to the following:

- if $A \le 292$, or if $A \le 3824$ and $R \le 0.67$, or if $R \le 0.25$, LDPC base graph 2 is used;
- otherwise, LDPC base graph 1 is used,

Source: https://www.etsi.org/deliver/etsi_ts/138200_138299/138212/15.03.00_60/ts_138212v150300p.pdf, 71.

65. The ’891 Accused Products include a data changing means for, if the

communication quality judged by the communication quality judging means does not satisfy a predetermined condition, making a predetermined change to the data to be transmitted represented by the symbol used in the judgment. According to 3GPP TS 38.212, the error detection provided by CRC includes judging the bit stream based on the payload size (A) and number of parity bits (L), and accordingly selecting the coding rate(R). Based on the values of A and R, a low-density parity check (LDPC) base graph is selected which is a method of coding and transmitting a message over a noisy transmission channel. For example, A ≤ 292 (i.e., predetermined condition), or if A ≤ 3824 and R ≤ 0.67, or if R ≤ 0.25, LDPC base graph 2 is used. Otherwise, LDPC base graph 1 is used. LDPC codes are used to correct the errors, and, therefore, correcting the error corresponds to changing data.

5 General procedures

Data and control streams from/to MAC layer are encoded /decoded to offer transport and control services over the radio transmission link. Channel coding scheme is a combination of error detection, error correcting, rate matching, interleaving and transport channel or control information mapping onto/splitting from physical channels.

5.1 CRC calculation

Denote the input bits to the CRC computation by $a_0, a_1, a_2, a_3, ..., a_{A-1}$, and the parity bits by $p_0, p_1, p_2, p_3, ..., p_{L-1}$, where A is the size of the input sequence and L is the number of parity bits. The parity bits are generated by one of the following cyclic generator polynomials:

- $g_{CRC24A}(D) = [D^{24} + D^{23} + D^{18} + D^{17} + D^{14} + D^{11} + D^{10} + D^{7} + D^{6} + D^{5} + D^{4} + D^{3} + D + 1]$ for a CRC length $L = 24$;
- $g_{CRC24B}(D) = [D^{24} + D^{23} + D^{6} + D^{5} + D + 1]$ for a CRC length $L = 24$;
- $g_{CRC24C}(D) = [D^{24} + D^{23} + D^{21} + D^{20} + D^{17} + D^{15} + D^{13} + D^{12} + D^{8} + D^{4} + D^{2} + D + 1]$ for a CRC length $L = 24$;
- $g_{CRC16}(D) = [D^{16} + D^{12} + D^{5} + 1]$ for a CRC length $L = 16$;
- $g_{CRC11}(D) = [D^{11} + D^{10} + D^{9} + D^{5} + 1]$ for a CRC length $L = 11$;
- $g_{CRC6}(D) = [D^{6} + D^{5} + 1]$ for a CRC length $L = 6$.

The encoding is performed in a systematic form, which means that in GF(2), the polynomial:

$$a_0 D^{A+L-1} + a_1 D^{A+L-2} + ... + a_{A-1} D^{L} + p_0 D^{L-1} + p_1 D^{L-2} + ... + p_{L-2} D^{1} + p_{L-1}$$

Source:
https://www.etsi.org/deliver/etsi_ts/138200_138299/138212/15.03.00_60/ts_138212v150300p.pdf, 9.

7.2.2 LDPC base graph selection

For initial transmission of a transport block with coding rate R indicated by the MCS index according to Clause 5.1.3.1 in [6, TS 38.214] and subsequent re-transmission of the same transport block, each code block of the transport block is encoded with either LDPC base graph 1 or 2 according to the following:

- if $A \leq 292$, or if $A \leq 3824$ and $R \leq 0.67$, or if $R \leq 0.25$, LDPC base graph 2 is used;
- otherwise, LDPC base graph 1 is used,

Source: https://www.etsi.org/deliver/etsi_ts/138200_138299/138212/15.03.00_60/ts_138212v150300p.pdf, 70.

LDPC

LDPC stands for "Low-Density Parity-Check". It's a method used in communication systems for encoding and decoding data to detect and correct errors. LDPC codes are a type of linear error-correcting code that has found extensive application in modern communication technologies. LDPC codes are a powerful and efficient method for error correction in digital communication systems, enabling reliable data transmission even in challenging and noisy environments.

Here are some key points about LDPC:

- **Error Correction**: LDPC codes are particularly effective for correcting data transmission errors in noisy communication channels. They are used in scenarios where accurate data transmission is critical, such as in satellite communication, wireless networks, and data storage.
- **Sparse Parity-Check Matrix**: The 'low-density' in LDPC refers to the parity-check matrix used in these codes, which has a relatively small number of non-zero elements. This sparsity is what makes LDPC codes computationally efficient for error correction.

Source: https://www.sharetechnote.com/html/5G/5G_LDPC.html.

66. The ’891 Accused Products include wherein at least a portion of a bit string is distinguished as a protected portion, the bit string constituting data to be transmitted represented by the sequence of symbols, and at least a portion of the symbol that belongs to the sequence of symbols contains a bit belonging to the protected portion and a redundant bit having a predetermined value. According to 3GPP TS 38.212, each code block of the transport block is encoded with either LDPC base graph 1 or 2 based on the payload value (A) and coding rate (R) as shown below. The entire transport block is used to calculate the CRC parity bits using the generator polynomial which depends upon the number of the parity bits and payload size. The coded bit stream (i.e., a bit string) with CRC bits attached to it, is then divided into multiple blocks comprising of useful bits (i.e., a protected portion) and redundant bits represented as Cr0, Cr1…, Cr(kr-1) where r is the code block number and Kr is the number of bits for code block number r. Thus, each code segment has some useful bits and some parity/CRC bits.

7.2.2 LDPC base graph selection

For initial transmission of a transport block with coding rate R indicated by the MCS index according to Clause 5.1.3.1 in [6, TS 38.214] and subsequent re-transmission of the same transport block, each code block of the transport block is encoded with either LDPC base graph 1 or 2 according to the following:

- if $A \le 292$, or if $A \le 3824$ and $R \le 0.67$, or if $R \le 0.25$, LDPC base graph 2 is used;
- otherwise, LDPC base graph 1 is used,

Source: https://www.etsi.org/deliver/etsi_ts/138200_138299/138212/15.03.00_60/ts_138212v150300p.pdf, 71.

6.2.3 Code block segmentation and code block CRC attachment

The bits input to the code block segmentation are denoted by $b_0, b_1, b_2, b_3, ..., b_{B-1}$ where B is the number of bits in the transport block (including CRC).

Code block segmentation and code block CRC attachment are performed according to Subclause 5.2.2.

The bits after code block segmentation are denoted by $c_{r0}, c_{r1}, c_{r2}, c_{r3}, ..., c_{r(K_r-1)}$, where r is the code block number and K_r is the number of bits for code block number r according to Subclause 5.2.2.

Source: https://www.etsi.org/deliver/etsi_ts/138200_138299/138212/15.03.00_60/ts_138212v150300p.pdf, 71.

5G Modulation and Coding Scheme

For any communication technology, Modulation and Coding Scheme (MCS) defines the numbers of useful bits which can carried by one symbol. In contrast with 5G or 4G, a symbol is defined as Resource Element **(RE)** and MCS defined as how many useful bits can be transmitted per Resource Element (RE) . **MCS** depends on radio signal quality in wireless link, better quality the higher MCS and the more useful bits can be transmitted with in a symbol and bad signal quality result in lower MCS means less useful data can be transmitted with in a symbol.

In other words, we can say MCS depends Blocker Error Rate **(BLER)**. Typically there is a BLER threshold defined that equal to 10%. To maintain BLER not more than this value in varying radio condition Modulation and Coding Scheme (MCS) is allocated by **gNB** using link adaptation algorithm. The allocated **MCS** is signalled to the UE using DCI over PDCCH channel e.g. **DCI 1_0**, **DCI 1_1**

A **MCS** basically defines the following two aspects:

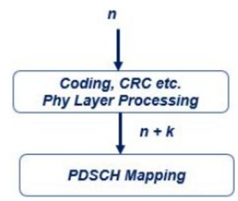
- **Modulation**
- **Code rate**

Code Rate

Code rate can be defined as the **ratio** between useful bit and total transmitted bit (Useful + Redundant Bits). These Redundant bits are added for Forward Error Correction **(FEC)**. In other words we can it is the ratio between the number of information bits at the top of the Physical layer and the number of bits which are mapped to **PDSCH** at the bottom of the **Physical layer**. We can also say, it a measure of the redundancy which is added by the Physical layer. A low coding rate corresponds to increased **redundancy**.

$$\text{Code Rate } (R) = \frac{n}{n+k}$$

$$\text{Code Rate } (R) = \frac{120}{1024} = 0.117$$



MCS Index I_{MCS}	Modulation Order Q_m	Target code Rate *R* x [1024]	Spectral efficiency
0	2	120	0.2344
1	2	157	0.3066
2	2	193	0.3770
3	2	251	0.4902
4	2	308	0.6016
5	2	379	0.7402

Source: https://www.techplayon.com/5g-nr-modulation-and-coding-scheme-modulation-and-code-rate/

67. The ’891 Accused Products include wherein the communication quality judging means identifies the number of redundant bits having the predetermined value or the number of redundant bits missing the predetermined value among the redundant bits contained in the symbol that contains a bit belonging to the protected portion, and judges the communication quality of the transmission channel based on the identified result. According to 3GPP TS 38.212, the bit stream is divided into multiple segments with each segment comprised of useful bits and redundant bits. Further, CRC calculations are done using the generator polynomial shown below. During the calculations, in the context of the applied input bits and a generator polynomial, redundant bits are predetermined bits. These attached CRC bits can be used for purpose error detection, which is representative of communication quality. Also, the amount of these redundant bits indicates the measure of redundancy which in turn indicates the corresponding coding rate. A low coding rate corresponds to increased redundancy and subsequent re-transmission. Hence, the channel quality can be judged by the CRC check using the generator polynomial. The communication quality judging step is done on the basis of the redundant bits contained in the symbol that contains a

protected portion. The protected data is the voice data with some special features, such as sound pressure, pitch, etc., and has the predetermined value "1". Therefore, the entire voice data coded by LDPC in the 5G NR case must include protected data. In LDPC-based coding, the code block segments are created before applying the CRC calculation and detecting the error, and therfore the code block segment contains that protected data (i.e., ...the redundant bits contained in the symbol that contains a bit belonging to the protected portion").

5 General procedures

Data and control streams from/to MAC layer are encoded /decoded to offer transport and control services over the radio transmission link. Channel coding scheme is a combination of error detection, error correcting, rate matching, interleaving and transport channel or control information mapping onto/splitting from physical channels.

5.1 CRC calculation

Denote the input bits to the CRC computation by $a_0, a_1, a_2, a_3, ..., a_{A-1}$, and the parity bits by $p_0, p_1, p_2, p_3, ..., p_{L-1}$, where A is the size of the input sequence and L is the number of parity bits. The parity bits are generated by one of the following cyclic generator polynomials:

- $g_{CRC24A}(D) = [D^{24} + D^{23} + D^{18} + D^{17} + D^{14} + D^{11} + D^{10} + D^{7} + D^{6} + D^{5} + D^{4} + D^{3} + D + 1]$ for a CRC length $L = 24$;
- $g_{CRC24B}(D) = [D^{24} + D^{23} + D^{6} + D^{5} + D + 1]$ for a CRC length $L = 24$;
- $g_{CRC24C}(D) = [D^{24} + D^{23} + D^{21} + D^{20} + D^{17} + D^{15} + D^{13} + D^{12} + D^{8} + D^{4} + D^{2} + D + 1]$ for a CRC length $L = 24$;
- $g_{CRC16}(D) = [D^{16} + D^{12} + D^{5} + 1]$ for a CRC length $L = 16$;
- $g_{CRC11}(D) = [D^{11} + D^{10} + D^{9} + D^{5} + 1]$ for a CRC length $L = 11$;
- $g_{CRC6}(D) = [D^{6} + D^{5} + 1]$ for a CRC length $L = 6$.

The encoding is performed in a systematic form, which means that in GF(2), the polynomial:

$$a_0 D^{A+L-1} + a_1 D^{A+L-2} + ... + a_{A-1} D^{L} + p_0 D^{L-1} + p_1 D^{L-2} + ... + p_{L-2} D^{1} + p_{L-1}$$

Source:
https://www.etsi.org/deliver/etsi_ts/138200_138299/138212/15.03.00_60/ts_138212v150300p.pdf, 9.

7.2.2 LDPC base graph selection

For initial transmission of a transport block with coding rate R indicated by the MCS index according to Clause 5.1.3.1 in [6, TS 38.214] and subsequent re-transmission of the same transport block, each code block of the transport block is encoded with either LDPC base graph 1 or 2 according to the following:

- if $A \leq 292$, or if $A \leq 3824$ and $R \leq 0.67$, or if $R \leq 0.25$, LDPC base graph 2 is used;
- otherwise, LDPC base graph 1 is used,

Source:
https://www.etsi.org/deliver/etsi_ts/138200_138299/138212/15.03.00_60/ts_138212v150300p.pdf, 71.

6.2.3 Code block segmentation and code block CRC attachment

The bits input to the code block segmentation are denoted by $b_0, b_1, b_2, b_3, ..., b_{B-1}$ where B is the number of bits in the transport block (including CRC).

Code block segmentation and code block CRC attachment are performed according to Subclause 5.2.2.

The bits after code block segmentation are denoted by $c_{r0}, c_{r1}, c_{r2}, c_{r3}, ..., c_{r(K_r-1)}$, where r is the code block number and K_r is the number of bits for code block number r according to Subclause 5.2.2.

Source: https://www.etsi.org/deliver/etsi_ts/138200_138299/138212/15.03.00_60/ts_138212v150300p.pdf, 71.

5G Modulation and Coding Scheme

For any communication technology, Modulation and Coding Scheme (MCS) defines the numbers of useful bits which can carried by one symbol. In contrast with 5G or 4G, a symbol is defined as Resource Element **(RE)** and MCS defined as how many useful bits can be transmitted per Resource Element (RE) . **MCS** depends on radio signal quality in wireless link, better quality the higher MCS and the more useful bits can be transmitted with in a symbol and bad signal quality result in lower MCS means less useful data can be transmitted with in a symbol.

In other words, we can say MCS depends Blocker Error Rate **(BLER)**. Typically there is a BLER threshold defined that equal to 10%. To maintain BLER not more than this value in varying radio condition Modulation and Coding Scheme (MCS) is allocated by **gNB** using link adaptation algorithm. The allocated **MCS** is signalled to the UE using DCI over PDCCH channel e.g. **DCI 1_0**, **DCI 1_1**

A **MCS** basically defines the following two aspects:

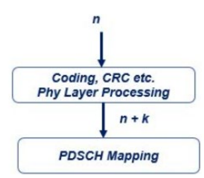
- **Modulation**
- **Code rate**

Code Rate

Code rate can be defined as the **ratio** between useful bit and total transmitted bit (Useful + Redundant Bits). These Redundant bits are added for Forward Error Correction **(FEC)**. In other words we can it is the ratio between the number of information bits at the top of the Physical layer and the number of bits which are mapped to **PDSCH** at the bottom of the **Physical layer**. We can also say, it a measure of the redundancy which is added by the Physical layer. A low coding rate corresponds to increased **redundancy**.

$$\text{Code Rate } (R) = \frac{n}{n+k}$$

$$\text{Code Rate } (R) = \frac{120}{1024} = 0.117$$



MCS Index I_{MCS}	Modulation Order Q_m	Target code Rate R x [1024]	Spectral efficiency
0	2	120	0.2344
1	2	157	0.3066
2	2	193	0.3770
3	2	251	0.4902
4	2	308	0.6016
5	2	379	0.7402

Source: https://www.techplayon.com/5g-nr-modulation-and-coding-scheme-modulation-and-code-rate/.

68. Apple has indirectly infringed and continues to indirectly infringe one or more claims of the ’891 Patent, as provided by 35 U.S.C. § 271(b), by inducing infringement by others,

such as Apple's customers and end-users, in this Judicial District and elsewhere in the United States. For example, Apple's customers and end-users directly infringe, either literally or under the doctrine of equivalents, through their use of the inventions claimed in the '891 Patent. Apple induces this direct infringement through its affirmative acts of manufacturing, selling, distributing, and/or otherwise making available the '891 Accused Products, and providing instructions, documentation, and other information to customers and end-users suggesting that they use the '891 Accused Products in an infringing manner, including technical support, marketing, product manuals, advertisements, and online documentation. Because of Apple's inducement, Apple's customers and end-users use the '891 Accused Products in a way Apple intends and they directly infringe the '891 Patent. Apple performs these affirmative acts with knowledge of the '891 Patent and with the intent, or willful blindness, that the induced acts directly infringe the '891 Patent.

69. Apple has indirectly infringed and continues to indirectly infringe one or more claims of the '891 Patent, as provided by 35 U.S.C. § 271(c), by contributing to direct infringement by others, such as customers and end-users, in this Judicial District and elsewhere in the United States. Apple's affirmative acts of selling and offering to sell the '891 Accused Products in this Judicial District and elsewhere in the United States and causing the '891 Accused Products to be manufactured, used, sold, and offered for sale contribute to others' use and manufacture of the '891 Accused Products, such that the '891 Patent is directly infringed by others. The accused components within the '891 Accused Products including, but not limited to, software manufactured by Apple, are material to the invention of the '891 Patent, are not staple articles or commodities of commerce, have no substantial non-infringing uses, and are known by Apple to be especially made or adapted for use in the infringement of the '891 Patent. Apple performs these affirmative acts with knowledge of the '891 Patent and with intent, or willful blindness, that they

cause the direct infringement of the ’891 Patent.

70. Apple has had actual notice of the ’891 Patent from either related prior litigations accusing products with similar AV1 functionalities involving direct competitors of Defendant and/or from its knowledge of ACT’s patent portfolio in *Advanced Coding Technologies, LLC v. Apple, Inc.*, Case No. 2:24-cv-00572-JRG (E.D. Tex. July 22, 2024).

71. ACT has suffered damages as a result of Defendant’s direct and indirect infringement of the ’891 Patent in an amount to be proved at trial.

DEMAND FOR JURY TRIAL

Plaintiff hereby demands a jury for all issues so triable.

PRAYER FOR RELIEF

WHEREFORE, ACT prays for relief against Defendant as follows:

a. Entry of judgment declaring that Defendant has directly and/or indirectly infringed one or more claims of each of the Patents-in-Suit;

b. An order pursuant to 35 U.S.C. § 283 permanently enjoining Defendant, its officers, agents, servants, employees, attorneys, and those persons in active concert or participation with them, from further acts of infringement of the Patents-in-Suit;

c. An order awarding damages sufficient to compensate ACT for Defendant’s infringement of the Patents-in-Suit, but in no event less than a reasonable royalty, together with interest and costs;

d. Entry of judgment declaring that Defendant’s infringement has been willful and awarding ACT treble damages pursuant to 35 U.S.C. § 284; and

e. Entry of judgment declaring that this case is exceptional and awarding ACT its costs and reasonable attorney fees under 35 U.S.C. § 285; and

f. Such other and further relief as the Court deems just and proper.

Dated: August 20, 2024

Respectfully submitted,

/s/ Peter Lambrianakos
Alfred R. Fabricant
NY Bar No. 2219392
Email: ffabricant@fabricantllp.com
Peter Lambrianakos
NY Bar No. 2894392
Email: plambrianakos@fabricantllp.com
Vincent J. Rubino, III
NY Bar No. 4557435
Email: vrubino@fabricantllp.com
Joseph M. Mercadante
NY Bar No. 4784930
Email: jmercadante@fabricantllp.com
FABRICANT LLP
411 Theodore Fremd Avenue, Suite 206 South
Rye, New York 10580
Telephone: (212) 257-5797
Facsimile: (212) 257-5796

ATTORNEYS FOR PLAINTIFF
ADVANCED CODING TECHNOLOGIES, LLC