

EXHIBIT 2001

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

LENOVO (UNITED STATES), INC.,

Petitioner,

v.

TELEFONAKTIEBOLAGET LM ERICSSON,

Patent Owner.

Case No. IPR2025-00177

U.S. Patent 10,708,618

DECLARATION OF IAIN RICHARDSON

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	QUALIFICATIONS	2
III.	MATERIALS CONSIDERED	5
IV.	SCOPE OF OPINIONS	5
V.	RELATED ITC INVESTIGATION	6
VI.	LEGAL STANDARDS	6
A.	Presumption of Validity	7
B.	Claim Construction	7
C.	Obviousness	8
D.	Motivations to Combine	10
E.	Objective Indicia of Non-Obviousness	11
VII.	LEVEL OF ORDINARY SKILL IN THE ART	11
VIII.	CLAIM CONSTRUCTION	12
IX.	TECHNOLOGY BACKGROUND	13
A.	Introduction to Video Coding	13
B.	Introduction to Reference Picture Management	19
C.	Picture Types in H.264 and H.265	21
D.	H.264 Reference Picture Management	23
1.	Decoded Picture Buffer	26
2.	Sliding Window and MMCO Commands in H.264	31
3.	Common Problems With Using MMCO Commands and Sliding Window in H.264	36
4.	Reference Picture Lists	42
5.	Reference Picture List Reordering Commands	47
X.	U.S. PATENT NO. 10,708,618 (“’618 Patent”) (Ex-1001)	54
A.	Priority Date	54
B.	’618 Patent Overview	55
C.	The ’618 Patent’s Prosecution History	73
XI.	DEVELOPMENT OF THE H.265 STANDARD	77
XII.	ASSERTED REFERENCES	81
A.	Mulroy (Ex-1004)	81
B.	H.264 (Ex-1005)	86
C.	Raveendran (Ex-1006)	86
D.	Dr. Creusere’s Incorrect Characterizations of H.264 and Mulroy, Including His Examples	87

XIII.	THE CHALLENGED CLAIMS OF THE '618 PATENT ARE NOT OBVIOUS	93
A.	Petitioner's Ground 1 Fails Because Claims 1-7 Are Not Obvious In View Of Mulroy And H.264.....	93
1.	The prior art does not disclose or render obvious "buffer description information defining a plurality of reference pictures," as required by claims 1-7.	93
a.	Dr. Creusere does not apply the plain and ordinary meaning of "buffer description information defining a plurality of reference pictures"	93
b.	Mulroy in view of H.264 still requires the use of relative or implicit signaling instead of absolute and explicit signaling	99
2.	The prior art does not disclose or render obvious "a reference picture identifier that identifies a respective reference picture among the plurality of reference pictures, the respective reference picture comprising a decoding reference for the picture" as required by claims 1-7.	115
3.	The prior art does not disclose or render obvious "updating a decoded picture buffer based on the reference picture identifier," as required by claims 1-7.	118
4.	The prior art does not disclose or render obvious "i) a picture identifier identifying a respective reference picture among the plurality of reference pictures as the decoding reference for the picture and ii) a picture identifier identifying a respectiv reference picture among the plurality of reference pictures as a decoding reference for a subsequent, according to a decoding order, picture in the video," as required by claim 2.	121
B.	Petitioner's Ground 2 Fails Because Claims 8-19 Are Not Obvious In View Of Mulroy, H.264, and Raveendran	125
1.	The prior art does not disclose or render obvious "buffer description information defining a plurality of reference pictures," as required by claims 8-19.	125
2.	The prior art does not disclose or render obvious "a reference picture identifier that identifies a respective reference picture among the plurality of reference pictures, the respective reference picture comprising a decoding reference for the picture" as required by claims 8-19.	126
3.	The prior art does not disclose or render obvious "update a decoded picture buffer based on the reference picture identifier," as required by claims 8-19.	126
4.	The prior art does not disclose or render obvious "i) a picture identifier identifying a respective reference picture among the plurality of reference pictures as the decoding reference for the picture and ii) a picture identifier identifying a respectiv reference picture among the plurality of reference pictures as a decoding reference for a subsequent, according to a decoding order, picture in the video," as required by claim 9.	127

XIV. OBJECTIVE INDICIA OF NON-OBVIOUSNESS 127

XV. SUMMARY AND OTHER REMARKS..... 129

Table of Exhibits

Exhibit No.	Description
Ex-2002	<i>Curriculum Vitae</i> of Dr. Iain Richardson
Ex-2003	Excerpts of Richardson, I., “The H.264 Advanced Video Compression Standard,” 2d Ed. (2010)
Ex-2004	Sjöberg and Samuelsson, “Absolute Signaling of Reference Pictures,” JCTVC-F493 (July 18, 2011)
Ex-2005	Sjöberg et al., “JCT-VC AHG Report: Reference Picture Buffering and List Construction (AHG21),” JCTVC-G021 (November 21, 2011)
Ex-2006	Intentionally Omitted
Ex-2007	ITU-T, “SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of Audiovisual Services – Coding of Moving Video: High Efficiency Video Coding”, Recommendation ITU-T H.265 (November 2019)
Ex-2008	U.S. Appl. No. 61/503,019
Ex-2009	Complaint, <i>In the Matter of Certain Electronic Computing Devices, and Components and Modules Thereof</i> , 337-TA-1387 (U.S.I.T.C. December 12, 2023)
Ex-2010	Excerpts of Public Transcript from Evidentiary Hearing, <i>In the Matter of Certain Electronic Computing Devices, and Components and Modules Thereof</i> , 337-TA-1387 (U.S.I.T.C. Oct. 25, 2024) (Public Testimony of Patent Owner’s Expert, Dr. Richardson)
Ex-2011	Excerpts of Public Transcript from Evidentiary Hearing, <i>In the Matter of Certain Electronic Computing Devices, and Components and Modules Thereof</i> , 337-TA-1387 (U.S.I.T.C. Oct. 23-25, 2024) (Public Testimony of Petitioner’s Expert, Dr. Hu)
Ex-2012	Borgwardt, “Multi-picture Buffer Semantics for Interlaced Coding,” JVT-C049 (May 2002)
Ex-2013	Fröjd, “And the Emmy Goes to . . . HEVC,” Oct. 26, 2017 (available at https://www.ericsson.com/en/blog/2017/10/and-the-emmy-goes-to--hevc#:~:text=The%20Emmy%20for%20HEVC%20was,for%20more%20than%2020%20years)
Ex-2014	“HEVC Awarded Engineering Emmy,” Oct. 25, 2017 (available at https://www.ericsson.com/en/news/2017/10/hevc-awarded-engineering-emmy)

I. INTRODUCTION

1. My name is Dr. Iain Richardson, and I have been retained as a technical expert by counsel for Patent Owner Telefonaktiebolaget LM Ericsson (“Ericsson” or “Patent Owner”) to investigate and opine as to the validity of U.S. Patent No. 10,708,618 (“the ’618 Patent”) in view of Petitioner Lenovo Corp.’s (“Ppetitioner” or “Lenovo”) Petition for *Inter Partes* Review (the “Petition”) (Paper 1).

2. This declaration is based on information currently available to me. To the extent that additional information becomes available, I reserve the right to continue my investigation and study, and thus may expand or modify my opinions as my investigation and study continues. I may also supplement my opinions in response to any additional information that becomes available to me, any matters raised by Petitioner and/or opinions rendered by Petitioner’s experts, or in light of any relevant orders from the Board.

3. My opinions are based on my years of education, research and experience, as well as my investigation and study of relevant materials.

4. My company Vcodex BV is being compensated at the rate of \$1,000 per hour for my work in this case, including time spent testifying. I am also being reimbursed for reasonable fees and expenses, including hotel and travel expenses, incurred as a result of my work in this case. Neither I nor my company have received any additional compensation for my work in this Investigation. Neither Vcodex

BV's compensation nor any personal remuneration I receive from my company Vcodex BV or from my company Vcodex Ltd is tied in any way to the substance of my testimony or the outcome of this Investigation. I have no other financial interest in this Investigation or to the parties. I am available to offer these opinions at deposition and at trial if called upon to do so.

II. QUALIFICATIONS

5. I have summarized in this section my educational background, work experience, and other relevant qualifications. A true and correct copy of my curriculum vitae is attached as Ex-2002 to Patent Owner's Preliminary Response which includes a list of all other cases in which, during the previous four years, I have testified at trial or by deposition.

6. I am Director and co-founder of Vcodex BV and Vcodex Limited. I am an internationally recognized expert on video and image compression coding with 30 years of experience in the field of video, image, multimedia, and general-purpose data encoding, decoding, compression, decompression, and transmission.

7. I obtained a Masters of Engineering from Heriot-Watt University, Edinburgh, UK, in 1990. I received a Doctor of Philosophy (Ph.D.) in video compression coding from Robert Gordon University, Aberdeen, UK in 1999. I served as a full professor at Robert Gordon University for 17 years, where I taught undergraduate and graduate engineering courses on topics including signal

processing, communications engineering, image and video processing, and multimedia compression.

8. In 1993, I assumed a post as a Lecturer, then Reader and eventually a Full Professor in the field of image and video compression in the School of Engineering at Robert Gordon University. In 2009, I was awarded the position of Honorary Professor with the Robert Gordon University.

9. I am the author of five books and over 70 journal and conference papers on video compression, including two widely cited books on the H.264/MPEG-4 standards and a book that focuses on the H.265/HEVC standard. I am regularly invited to give lectures on video compression coding, including at the Smithsonian Institution and the US Patent and Trademark Office.

10. I am a member of Netherlands expert group NEN, the committee which represents the Netherlands in ISO/IEC MPEG standardization.

11. I have many years of experience in researching, developing and implementing video compression codecs in software and hardware. In 1995, during my employment at Robert Gordon University, I co-founded and became Technical Director, equivalent of Chief Technology Officer, at 4i2i Communications Limited. I held this position until 2000. In my role at 4i2i, I was responsible for and engaged in the design of software and hardware modules for compression and communication systems, including the design of video processing algorithms in software and

hardware for implementation on processing devices and the design of compression algorithms such as block-based transforms, run-length coding and variable-length coding in software and hardware. A modified version of one of my video compression hardware designs was successfully used by NASA to stream video from the International Space Station.

12. A particular focus of my research work has been the effect of transmission losses on compressed video. This was a key topic of my PhD Thesis (Richardson, 1999), and I have written many research papers on this topic.

13. In 2007, I founded Vcodex Limited, now Vcodex BV, and currently serve as the CEO, through which I provide technical and strategic advice to technology companies particularly relating to video, image, and data compression for streaming.

14. In 2009, I founded Onecodec Limited and served as CEO, leading the company's development of innovative video, image, and multimedia data compression and storage software and systems.

15. I am a named inventor on 11 patents or patent applications, including U.S. and non-U.S. patents and applications.

16. I believe I am qualified to render the opinions set forth here.

III. MATERIALS CONSIDERED

17. I have reviewed the '618 Patent, including the challenged claims, prosecution history, and relevant patent family. The '618 Patent was marked as Ex-1001, and its prosecution history was marked as Ex-1002.

18. I also reviewed the Petition, and each of Exhibits 1001-1016 attached to the Petition.

19. In forming my opinions, I have considered the materials listed above and any other documents cited in this declaration. I have also relied on my own education, knowledge, and experience in the relevant art.

20. I have also considered the understanding of a person of ordinary skill around the time of the invention of the '618 Patent.

IV. SCOPE OF OPINIONS

21. I set forth my opinions throughout this declaration.

22. I understand that Petitioner argues that the challenged claims are obvious over Mulroy and H.264 (Ground 1) and Mulroy, H.264, and Raveendran (Ground 2).

23. I disagree. It is my opinion that the challenged claims of the '618 Patent are not obvious in view of the asserted references, alone or in combination, for the reasons discussed herein.

V. RELATED ITC INVESTIGATION

24. Patent Owner filed a complaint against Petitioner in the International Trade Commission (ITC) on December 12, 2023. Ex. 2009. Petitioner relied on the same primary references—Mulroy and the H.264 Standard—in that Investigation.

25. I served as an expert witness in that Investigation on behalf of Patent Owner. I provided opinions on these references and the '618 Patent in that Investigation, provided an expert report, and testified at the evidentiary hearing in October, 2024. My public testimony on the '618 Patent and related issues is attached as Ex-2010.

26. Dr. Creusere was not Lenovo's technical expert in the ITC Investigation. Lenovo retained Dr. Hu as a technical expert. Her public testimony on the '618 Patent and related issues is attached as Ex-2011.

27. I understand that a decision by the Administrative Law Judge (ALJ) is expected to issue before or on March 21, 2025.

VI. LEGAL STANDARDS

28. For purposes of this Declaration, I use the legal principles below as a guide in formulating my opinions. I am not an attorney, and I am not offering any opinions regarding legal matters; however, I have been informed by Ericsson's counsel of the legal principles relevant to the issues herein and have the following understanding.

A. Presumption of Validity

29. I understand that in deciding whether to issue a patent, the U.S. Patent and Trademark Office (“USPTO”) examines the patent specification, its claims, and relevant prior art references to determine whether the patent application and its claims meet the requirements for patentability. I understand that if the USPTO then issues the patent, it has determined that the claims are valid. I understand that the law recognizes that the USPTO possesses special expertise to conduct such determinations, and that, as a result, each claim of a patent issued by the USPTO is presumed valid by law.

30. I understand that this presumption of validity can be overcome in an *inter partes* review proceeding only if the party seeking to invalidate a claim proves invalidity by a preponderance of the evidence, which I understand to mean evidence that convinces you that it is more likely than not that the particular proposition is true. I understand that the preponderance of the evidence standard applies to all aspects of an allegation of anticipation or obviousness, including the prior art status of the relevant reference(s).

B. Claim Construction

31. I understand the first step in determining whether or not a patent claim is valid is to properly construe the claims. I also understand that, as a general matter, a claim should not be limited to a preferred embodiment, but that in certain cases,

the scope of the right to exclude may be limited by a narrow disclosure or by positions taken, such as by statements made during patent prosecution. I also understand the claims must be supported by the specification. Also, to the extent that a patent claims priority to an earlier filed application, the claims must be supported by the disclosure in that application. For terms that have not been construed I understand that they should be afforded their plain and ordinary meaning to one of ordinary skill in the art.

32. I understand that the plain and ordinary meaning of a claim term is the meaning a person of ordinary skill in the art would have understood at the time of the effective date in view of the specification and the prosecution history.

C. Obviousness

33. I understand that a claimed invention is not patentable under 35 U.S.C. § 103 if the differences between the invention and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which the subject matter pertains. Obviousness, as I understand it, is based on the scope and content of the prior art, the differences between the prior art and the claim, the level of ordinary skill in the art, and objective indicia of non-obviousness to the extent they exist.

34. I understand that whether there are any relevant differences between the prior art and the claimed invention is to be analyzed from the view of a person

of ordinary skill in the art at the time of the invention. A person of ordinary skill in the art is a hypothetical person who is presumed to be aware of all of the relevant art at the time of the invention. The person of ordinary skill is not an automaton, and may be able to fit together the teachings of multiple patents employing ordinary creativity and the common sense that familiar items may have obvious uses in another context or beyond their primary purposes.

35. I understand that if a reference or proposed combination of references does not disclose or suggest all of the elements of a claim, the combination cannot render the claim obvious. I also understand that in order to combine prior art references to show obviousness, a person of ordinary skill in the art, without knowledge of the claimed invention, or without the use of hindsight, must have been motivated to combine the prior art by some suggestion or teaching in the prior art, the knowledge of one of skill in the art, or the nature of the problem to be solved.

36. An invention is obvious if a designer of ordinary skill in the art, facing the wide range of needs created by developments in the field, would have seen an obvious benefit to the solutions tried by the applicant. When there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, it would be obvious to a person of ordinary skill to try the known options. If a technique has been used to improve one device, and a person of ordinary skill in the art would recognize that it would improve similar devices in the

same way, using the technique would have been obvious. I understand, however, that routine experimentation does not necessarily preclude patentability.

37. I understand that obviousness of a patent claim cannot properly be established through hindsight, and that elements from different prior art references, or different embodiments of a single prior art reference, cannot be selected to create the claimed invention using the invention itself as a roadmap. I understand that the claimed invention as a whole must be compared to the prior art as a whole, and courts must avoid aggregating pieces of prior art through hindsight that would not have been combined absent the inventors' insight.

D. Motivations to Combine

38. I understand that in deciding obviousness, it is appropriate to consider whether some teaching, suggestion, or motivation would have led a person of ordinary skill in the art to combine the invalidating references. I also understand that it is appropriate to look to a variety of factors when evaluating reasons to combine references in an obviousness analysis, and that it is important to identify a reason that would have prompted a person of ordinary skill in the relevant field to combine the prior art elements in the manner claimed. I understand that in determining whether a combination of prior art references renders a claim obvious, it may be helpful to consider whether there is some teaching, suggestion, or motivation to combine the references and a reasonable expectation of success in

doing so. I understand, however, that the teaching, suggestion, or motivation to combine inquiry is not required and may not be relied upon in lieu of the obviousness analysis outlined above. I understand that a patent composed of several elements is not proved obvious merely by demonstrating that each of its elements was independently known in the prior art. I also understand that this is so because inventions in most, if not all, instances rely upon building blocks long since uncovered, and claimed discoveries almost of necessity will be combinations of what, in some sense, is already known.

E. Objective Indicia of Non-Obviousness

39. I understand that in evaluating whether patent claims are invalid as obvious, objective indicia of non-obviousness can be considered. Such objective indicia can include commercial success, long-felt but unsolved needs, copying, praise, unexpected results, industry acceptance, failure of others, and skepticism by experts.

VII. LEVEL OF ORDINARY SKILL IN THE ART

40. I understand that patent claims are to be interpreted from the viewpoint of one of ordinary skill in the relevant art. To determine the level of skill that would be “ordinary,” I understand that a person of ordinary skill (“POSITA”) must generally have the capability of understanding the general principles that are applicable to the pertinent art.

41. In my opinion, a POSITA would have had at least a bachelor's degree in electrical engineering, computer engineering, or a related field, with about two years of experience in video decoding or encoding technologies. More work or practical experience may qualify one not having the requisite education as a person with ordinary skill in the art while a higher level of education could offset less experience. I was at least a person of ordinary skill in the art as of the effective filing dates of the Asserted Patents under this definition.

42. I have reviewed Dr. Creusere's Declaration (Ex-1003). I understand that Dr. Creusere and I agree as to the definition of a POSITA. Ex-1003 at ¶ 35.

VIII. CLAIM CONSTRUCTION

43. I understand that unless claim terms are provided an express construction, the terms must be given their plain and ordinary meaning.

44. I understand that Petitioner applies the plain and ordinary meaning of all claim terms and contends that none of the terms require specific construction. Pet. at 17-18.

45. I also understand that in the ITC Investigation, Lenovo and Patent Owner dispute the meaning of the term "circuit," as reflected in Ex-1012. I understand that the ITC has not issued any claim construction for that term, but that for the purposes of this *inter partes* review, the Board need not construe the term.

46. Other than explaining the plain and ordinary meaning, which I do below, I do not offer an opinion on the correct construction of any disputed terms, because, in my opinion, the Challenged Claims are not taught or suggested by the prior art. I reserve the right to respond to any future claim construction argument asserted by Petitioner or Dr. Creusere.

47. Dr. Creusere states that the term “frame” and “picture” are used synonymously in the ’618 Patent. Ex-1003 at ¶ 96. While I agree that “frame” and “picture” can be used interchangeably in certain contexts, there are some instances where a distinction is required. A POSITA would have understood the terms “frame” and “picture” are not interchangeable throughout the H.264/AVC standard, for example. In H.264/AVC, a “frame” can be considered as a single progressive frame or a pair of interlaced fields and a “picture” can be a single frame or a single field. *Compare* Ex. 1005 at 7, § 3.55 *with id.* at 10, § 3.104. If a distinction between these two terms (frame and picture) is necessary, I state so in this Report.

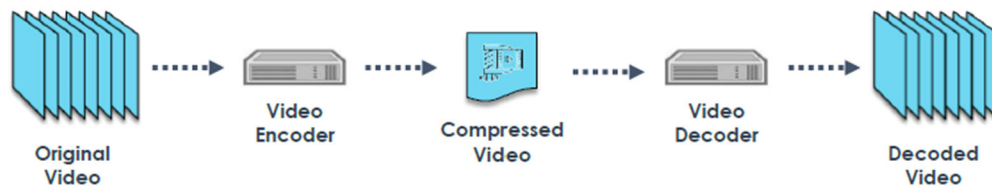
IX. TECHNOLOGY BACKGROUND

48. Below, I provide a brief overview of video coding technology and the relevant processes involved in encoding and decoding video.

A. Introduction to Video Coding

49. Video compression coding, often abbreviated to video coding, is the process of compressing (encoding) video prior to transmission or storage and

decompressing (decoding) compressed video so that it can be displayed. A video encoder converts digital video into a compressed form known as coded or compressed video, suitable for transmission or storage, and a video decoder converts coded or compressed video into uncompressed digital video. The pair of video encoder and video decoder is known as a video codec:



50. Uncompressed video typically consists of a series of frames, each of which is an image of the visual scene at a particular time instant. Each frame may be broken down into smaller regions called slices, and slices are further divided into regions called macroblocks. These macroblocks are made up of even smaller units called pixels.

51. Popular formats for uncompressed video include:

- Standard Definition (SD): Around 704 x 576 pixels per frame, displayed at 25 or 30 frames per second.
- High Definition (HD): 1920 x 1080 pixels per frame, displayed at 25 – 60 frames per second.
- Ultra-High Definition (UHD) or 4K: 3840 x 2160 pixels per frame, displayed at 50-60 frames per second.

52. Video coding is important because uncompressed video takes up a significant amount of storage space and would require an impractical amount of bandwidth to transmit in real time. For example, a 500Gbyte hard disk can only store around 40 minutes of uncompressed HD video. The same hard disk can store around 600 hours of compressed video, depending on the video content and the compression format.

53. Video data takes up the majority of storage and transmission capacity in today's consumer devices and networks, even in its compressed form. As display resolutions get bigger and consumers expect better video quality and higher resolution images when viewing stored video or video streamed over the internet, efficient and effective video compression continues to be of vital importance.

54. The performance of a video codec can be measured in at least three ways, namely compression, quality, and computation performance.

55. Compression relates to the extent to which data is reduced in size. This can be expressed as a ratio, *e.g.*, 100:1 compression means that the video, if stored in a file, occupies 1/100th of the space when compressed, compared with the uncompressed video. Alternatively, this can be expressed as a bitrate, *i.e.*, the transmission bitrate required to transmit the compressed video data in real time. For example, a typical compressed HD video clip might require between 2 and 4 Mbps (2-4 million bits per second) of transmission bitrate, to send the video in real time.

56. Quality relates to how the compressed and decompressed video appears to a user. Video compression is typically “lossy,” which means that the quality of the decoded video is not as good as the original, uncompressed video. For many consumer video applications, the aim is to provide the viewer with pleasing video quality, whether subjectively evaluated (i.e., how does a human observer rate the quality of the video) or objectively evaluated (i.e., an automatically calculated approximation of visual quality).

57. Computation relates to how much processing power is required to encode (compress) or decode (decompress) the video. Video coding uses complicated algorithms and methods that typically require significant computing time and resources and may require to be carried out in real time, for example to ensure continuous playback of decoded video.

58. These three aspects—compression, quality, and computation—are usually interrelated. For example, it may be possible to compress more and/or achieve higher video quality by using more computationally intensive compression processes or algorithms.

59. Many different techniques for implementing video coding designs have been researched and proposed. Each year, new and innovative technologies for improving compressions are described in many hundreds of research papers. However, in general, it is common for commercial video coding applications to use

a limited number of standardized techniques for video compression. Standardized video coding formats have several potential benefits when compared to non-standard formats including for example, interoperability. Standards for video coding simplify inter-operability between encoders and decoders from different manufacturers. This is critical in applications where each ‘end’ of the system (e.g., encoder and decoder) could be manufactured or provided by a different company. For example, a provider of streaming video content need not be (and is unlikely to be) the same company that manufactures the laptop or other device that receives, decodes and plays the streaming video content. As long as both the encoder and decoder operate in conformance with the same standard, any conforming decoder should be able to decode a compressed bitstream that was encoded in conformance with the standard.

60. Two specifications for video compression are H.264 (Advanced Video Coding or AVC) and H.265 (High Efficiency Video Coding or HEVC). ITU-T Recommendation H.264 is an industry standard for video compression that was first published in 2003. ITU-T Recommendation H.265, also referred to as High Efficiency Video Coding (HEVC) is an industry standard for video compression that was first published in 2013.

61. The H.265 standard can provide around twice the compression performance of the older H.264 standard. This means that an H.265 video encoder (i.e. a video encoder that produces a coded video bitstream that confirms to the H.265

standard) can compress video into a file that is about half the size of the file produced by an H.264 video encoder, for the same decoded video quality.

62. Both H.264 and H.265 were developed by committees of industry experts. H.264 was developed by the Joint Video Team (JVT) and H.265 was developed by the Joint Collaborative Team on Video Coding (JCT-VC), each comprising experts from the Video Coding Experts Group (a subgroup of ITU-T) and from the Moving Picture Experts Group (a sub-group of ISO/IEC). Both standards were co-published by ITU-T and ISO/IEC.

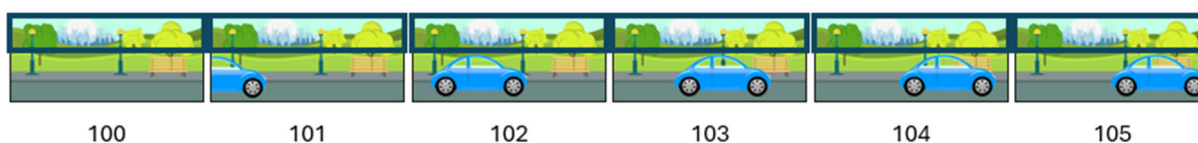
63. In each case (H.264 and H.265), the process of developing the standard took a number of years. Typically, such a process begins with an identification of a requirement for a new video coding standard and a Call for Proposals. Depending on the response to the Call for Proposals, the working group (JVT or JCT-VC) selects certain technologies to form the basis for further development (sometimes described as a Test Model) and then holds a series of meetings with the aim of continuing to refine the selected technologies. Once the Test Model reaches a certain level of maturity and performance, the working group proposes a Draft Standard which ultimately becomes a published international standard.

64. I provide an overview of reference picture management below, and I provide explanations of further aspects of video coding as necessary throughout this Declaration.

B. Introduction to Reference Picture Management

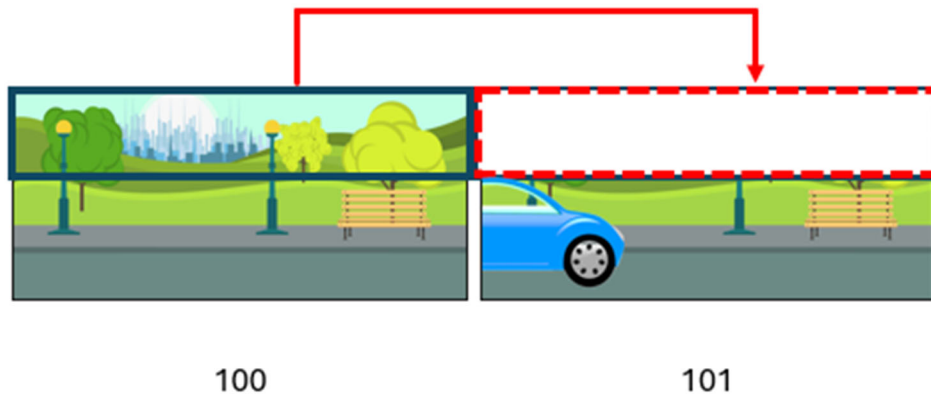
65. Modern video codecs rely on prediction techniques for effective compression. Often, pictures in a video sequence will repeat the same or similar content as other pictures in that sequence that occur at other moments in time. Inter prediction (illustrated below), or motion-compensated prediction, is a technique that takes advantage of this redundancy.

66. For example, the below illustration shows a sequence of pictures that make up a video of a car driving across the screen from left to right. The blue boxes at the top of each picture in this illustration show background content that is redundant between each picture. The numbers below each picture represent the picture numbers of each picture, which in this example are consistent with the order the pictures are displayed.



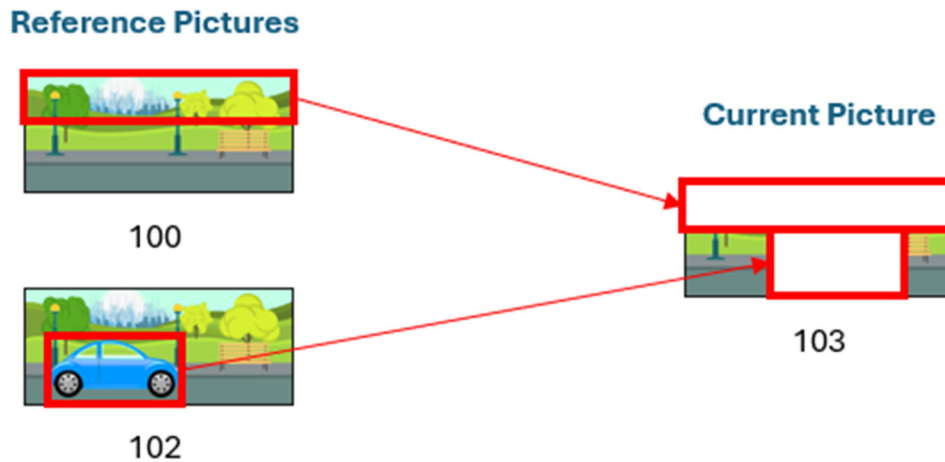
67. When using inter prediction, pictures are decoded using other pictures from the same video sequence as a reference. The pictures that are used as references for decoding other frames are called reference pictures. Reference pictures are a fundamental concept for inter prediction and modern video coding.

68. The following is a simple illustration to show how reference pictures are used.

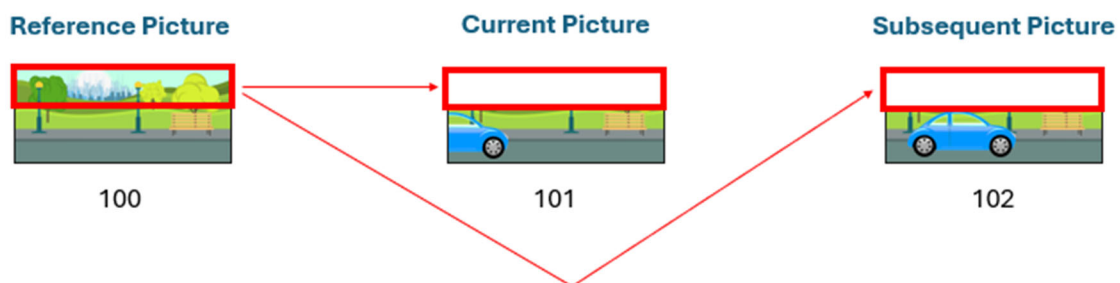


69. Here, Picture 100 is a reference picture for Picture 101. When encoding Picture 101, the encoder does not need to code the top part of Picture 101 into the bitstream. Instead, it will code information into the bitstream that instructs the decoder to copy the background part in blue of Picture 100 into Picture 101. This technique provides significant compression savings.

70. Multiple reference pictures can also be used for decoding a single picture. This is shown in the illustration below where Picture 103 uses Pictures 100 and 102 as reference pictures.



71. It is also common that the same reference picture can be used for decoding multiple pictures. This is shown in the illustration below where Pictures 101 and 102 both use Picture 100 as a reference picture.



C. Picture Types in H.264 and H.265

72. In the H.264 and H.265 standards, the category or type of picture dictates whether a reference picture should be used for decoding. In those standards,

there are three main categories of pictures: I, P, and B. An I picture is decoded without using any reference pictures. A special type of I picture, an Instantaneous Decoder Refresh (IDR), has the effect of resetting picture numbering and “clearing” the Decoded Picture Buffer (DPB), marking all pictures in the DPB as “unused for reference,” as I explain later. A P picture is one that uses reference pictures to be decoded, with each block in the picture capable of using prediction from one reference picture. A B picture is one that is decoded using reference pictures that may come before or after it in display order, with each block in the picture capable of using prediction from one or two reference pictures. The figure below illustrates the use of I, B, and P pictures.

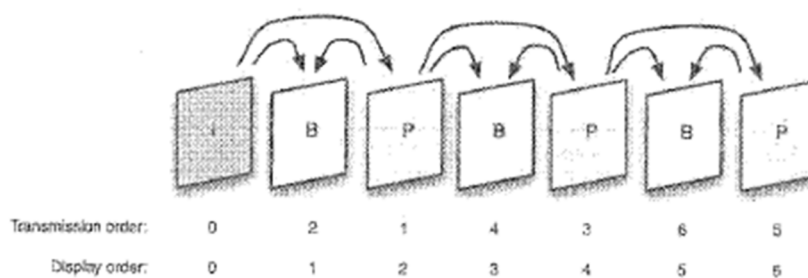


Figure 3.4 Display order: Type 0 example

Ex-2003 at 105.

73. A compression benefit of using multiple reference pictures is that the encoder may choose to predict each inter-coded block from one of multiple reference picture sources, enabling it to choose the source (reference picture) that provides the best prediction. At the same time, the compressed bitstream need only refer to certain previously coded reference pictures, not to every previously coded picture. The

bitstream will include one or more reference pictures and information that allows the decoder to predict the other pictures in the video stream from that reference picture(s). This can lead to significant bit savings, i.e. making the entire coding process more efficient in terms of compression.

74. The accuracy and reliability of decoding relies on using the correct reference picture(s) for decoding each picture. If the decoder uses the wrong reference picture or if the required reference picture is not available, then the picture that is being decoded (which is referred to as the “current picture”) cannot be correctly decoded. This type of error is especially likely to occur if a coded picture or coded information associated with the picture is “lost,” meaning that it somehow did not make it to the decoder or the picture got dropped by the decoder before it was done being needed as a reference picture. Therefore, the ways in which a video codec manages reference pictures is critical to the accuracy and reliability of the codec. Transmission errors such as bit errors or lost network packets are important practical concerns that can have a serious impact on decoded video quality. Ex-2003 at 247-8; Ex-1001 at 2:56-3:6.

D. H.264 Reference Picture Management

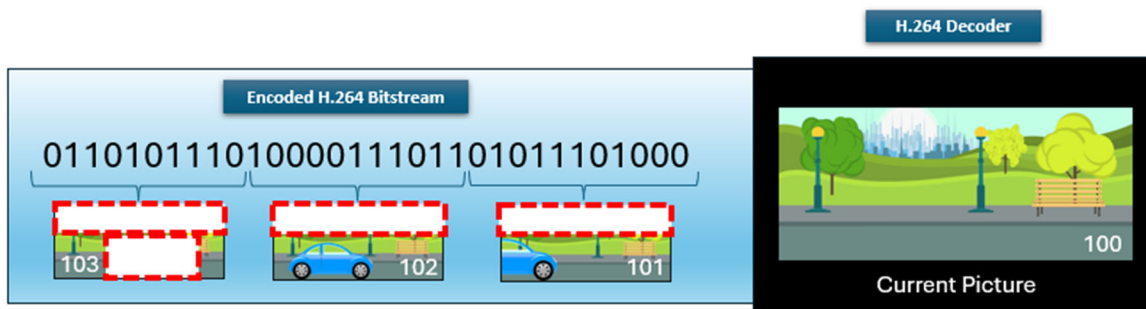
75. The H.264 standard defines syntax and semantics for encoded data in a video bitstream and specifies certain decoding processes. Ex-1005 at §§ 7-8. An H.264 bitstream consists of a series of encoded syntax elements, where each syntax

element represents an item of data such as coded representations of pixel data, information related to prediction (such as reference picture indices, discussed below) and other parameters. Syntax elements may be encoded using fixed length codewords (each containing a fixed number of bits), variable length codewords (each containing a number of bits that may vary depending on the value of the syntax elements) and (if the optional Context Adaptive Binary Arithmetic Coding or CABAC mode is selected) binary arithmetic encoding.

76. The H.264 standard specifies the way or ways in which each syntax element may be encoded. For example, the syntax element denoted `abs_diff_pic_num_minus1` (sent at most once per slice and discussed below) is encoded as a variable length codeword, where the length of the codeword depends on the value of the syntax element. As another example, the syntax element denoted `ref_idx_10` (sent once per inter-predicted block of pixels and discussed below) is encoded as a variable length codeword (if CABAC mode is not selected) or using binary arithmetic coding (if CABAC mode is selected). When this syntax element is coded using binary arithmetic coding, the value of the syntax element is first converted into a binary string known as a binarization (a variable-length string of binary digits or “bins”) and then the bins of the binarization are arithmetically coded. (H.264 section 9).

77. The following describes how reference picture management operates in H.264 using a series of demonstrative illustrations that are similar to the demonstratives that I presented in the ITC.

78. First, I have created the illustration below that shows an example of a bitstream (for ease of depiction, illustrated as a series of binary 0s and 1s) being received by the decoder. In this example, there are four Pictures in the bitstream.



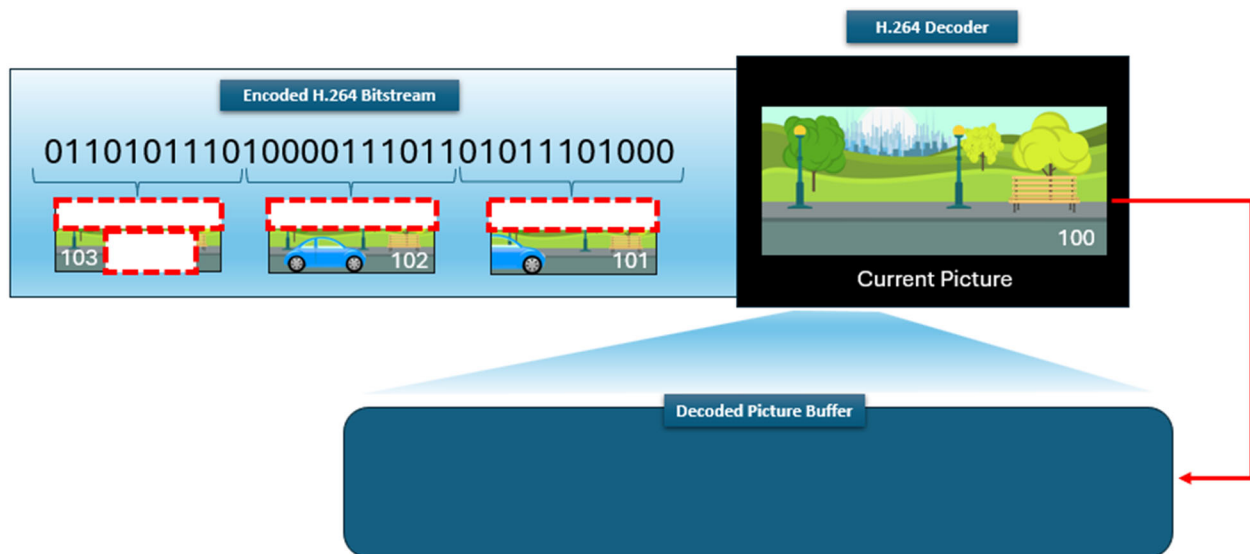
79. The black box illustrates the H.264 decoder, and the example illustrates that Picture 100 is about to be decoded. This illustration is intended to show the point in time immediately before Picture 100 is to be decoded. Picture 100 does not use any reference pictures and is an I (or IDR) picture.

80. The blue box to the left illustrates the encoded bitstream. The 0s and 1s are the binary values that represent the pictures that make up the video. In this example, Pictures 101, 102, and 103 are to be decoded next. These pictures are

illustrated with red-dashed boxes representing the portions of the pictures that will be predicted using reference pictures.

1. *Decoded Picture Buffer*

81. The following illustration adds a Decoded Picture Buffer (DPB) as a dark blue box.

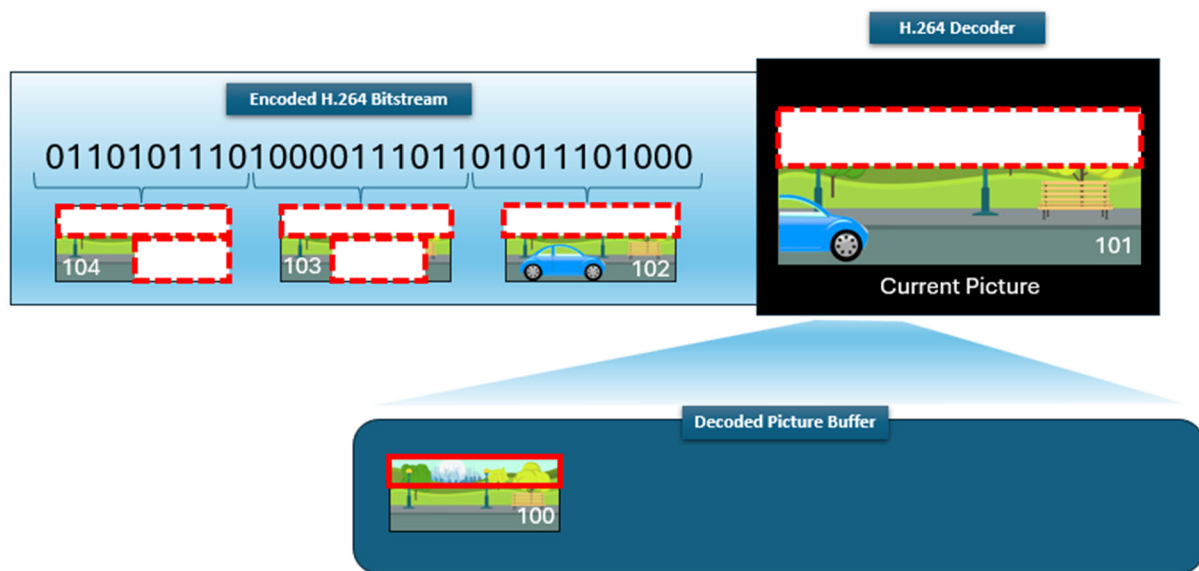


82. In H.264, the decoder maintains a DPB. The DPB stores previously decoded pictures that may be used as reference pictures for future pictures. Ex. 2003 at 104, 106.

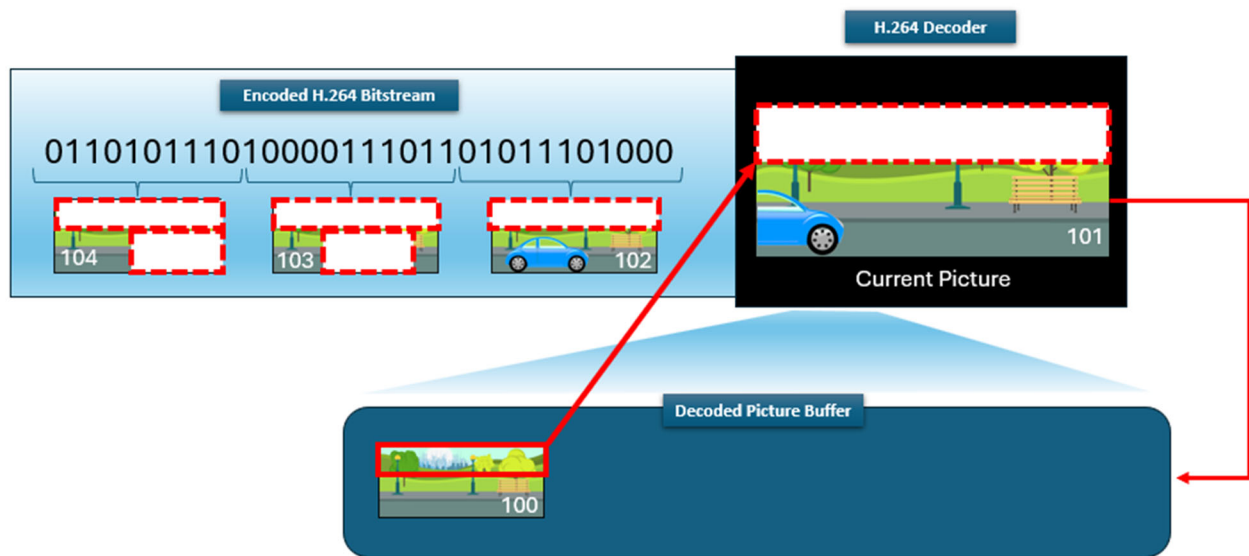
83. When the decoder begins processing a bitstream, there are no pictures in the DPB. Pictures may be added to the DPB as each picture in the bitstream is decoded. In the illustration above, the DPB is empty because no pictures have been

decoded. The red lines and arrow to the right illustrate that, as a Picture is decoded, it is stored in the DPB.

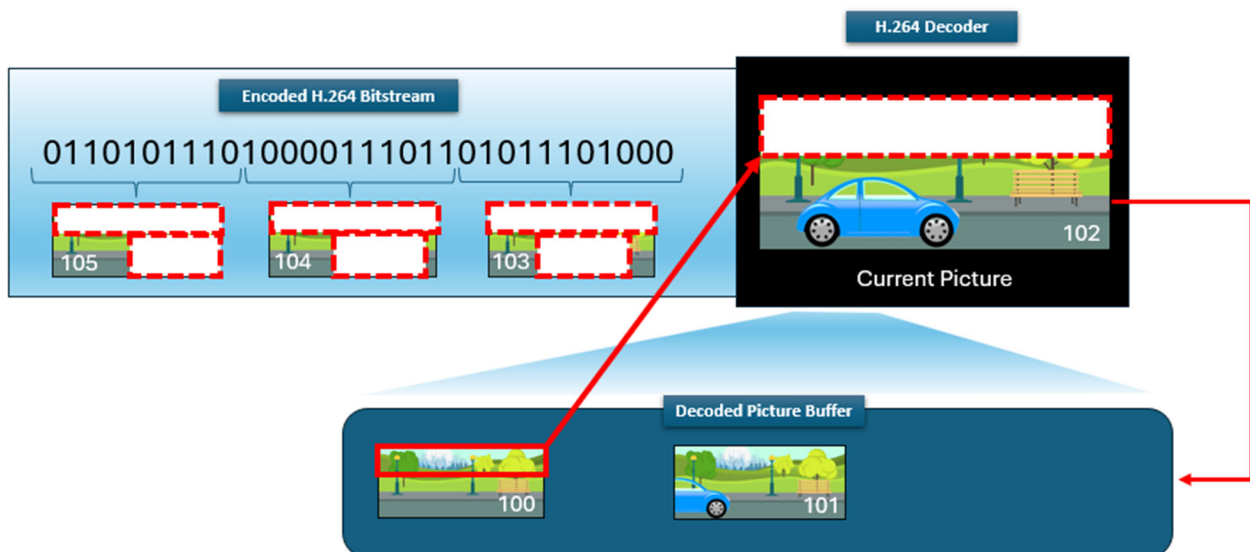
84. For example, after Picture 100 is decoded, it is sent to the DPB, and the following pictures move one step closer to decoding. At that point, Picture 101 is decoded, and Picture 100 exists in the DPB. This is illustrated below.



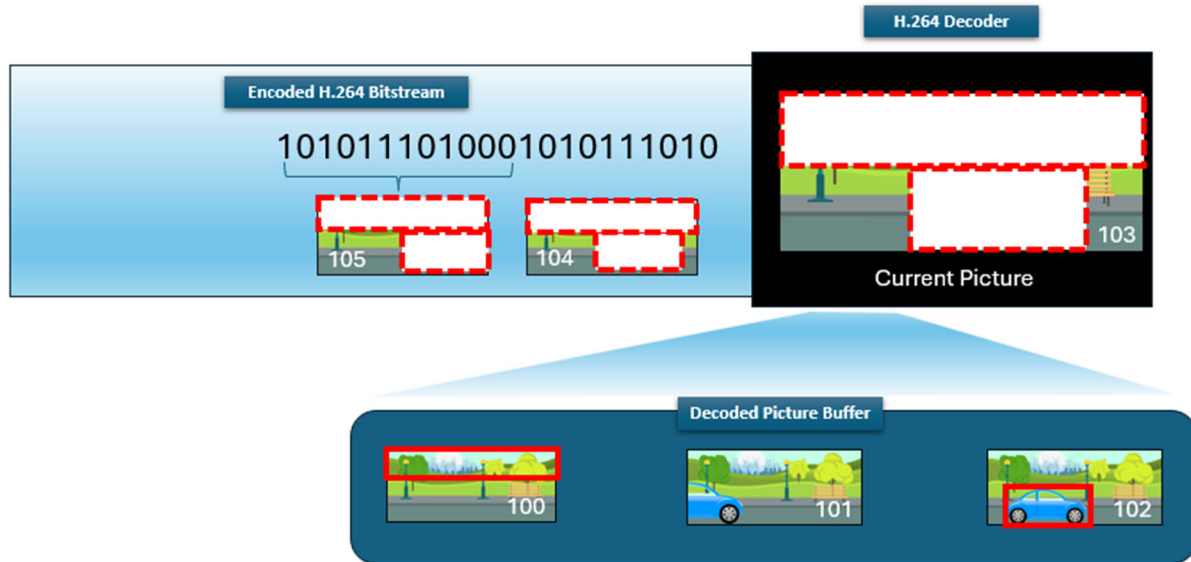
85. Picture 100 can now be used as a reference picture, for example to decode Picture 101. The decoder can predict Picture 101, in this example by copying areas of Picture 100 (highlighted with a red box in the illustration below) to Picture 101 as part of the decoding process.



86. As shown below, once Picture 101 is decoded, it too is stored in the DPB. Picture 100 remains in the DPB and in this example is used as a reference picture for decoding Picture 102, which is the next picture in the bitstream. This is illustrated below.



87. Once Picture 102 is decoded, it is stored in the DPB and can be used as a reference picture for decoding further pictures in the bitstream.

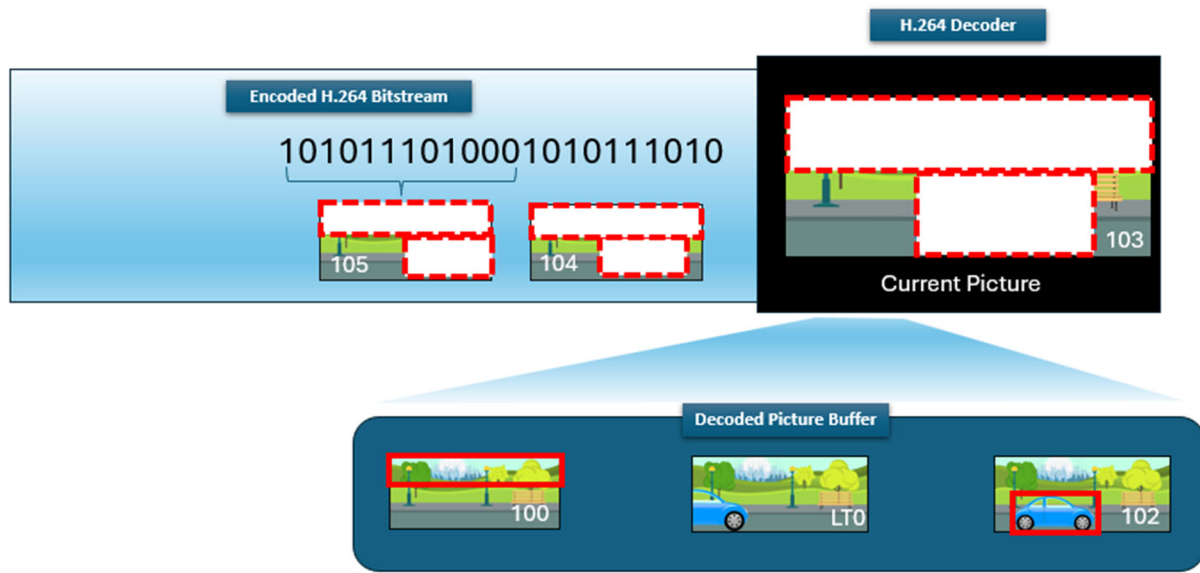


88. This process would continue until all pictures in the bitstream are decoded. It is important to note that the DPB will typically be implemented such that it has limited capacity. As might be expected, storage of pictures in the DPB is memory intensive, and thus designing the DPB to have a capacity is critical to the performance of the codec. To illustrate this point, in the example above, the DPB capacity is 3 pictures.

89. To assist the codec in maintaining the DPB in H.264, the pictures in the DPB are identified as short-term pictures or long-term pictures, with the identification influencing how long each picture remains in the DPB. Ex. 2003 at 106. A short-term picture will typically not remain in the DPB for as long as a long-

term picture, but may still be used as a reference picture for multiple pictures. A long-term picture may remain in the DPB for longer than a short-term picture and may be used as a reference picture for multiple pictures.

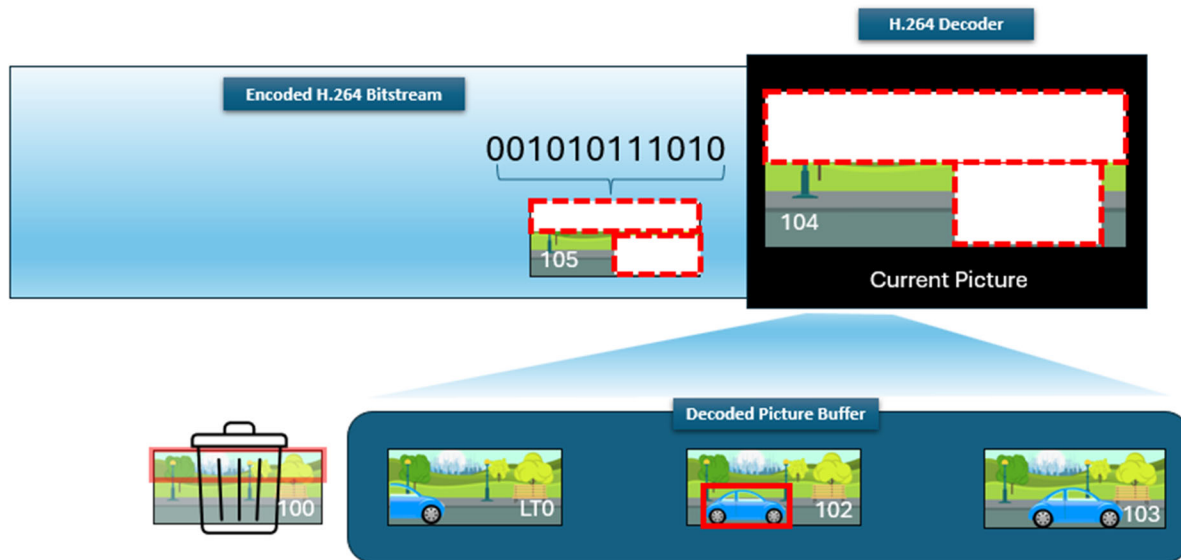
90. In H.264, a short-term picture is indexed by the picture number PicNum (which relates to the encoding and decoding order of frames) and a long-term picture in the DPB is indexed by LongTermFrameIndex or LongTermPicNum (a long-term frame index, not based on coding order or display order of frames) that can be assigned through MMCO commands, as I explain later in Paragraphs 98-103 and 108-12. In our example above, all three pictures in the DPB are short-term pictures and are indexed by picture number (100, 101, and 102). If Picture 101 were a long-term picture instead, it would be given an index “slot” denoted by LongTermFrameIndex, such as Index 0. This alternative example would look like this, where “LT” stands for long-term:



2. *Sliding Window and MMCO Commands in H.264*

91. The DPB should contain every reference picture needed to decode a given picture. There is a finite amount of memory (i.e. storage space) available at the decoder, and thus the decoder cannot maintain every previously decoded picture in its DPB.

92. In our illustration above, for example, the maximum number of reference pictures in the DPB is three. Once a fourth reference picture is decoded, the decoder must remove at least one picture from the DPB.



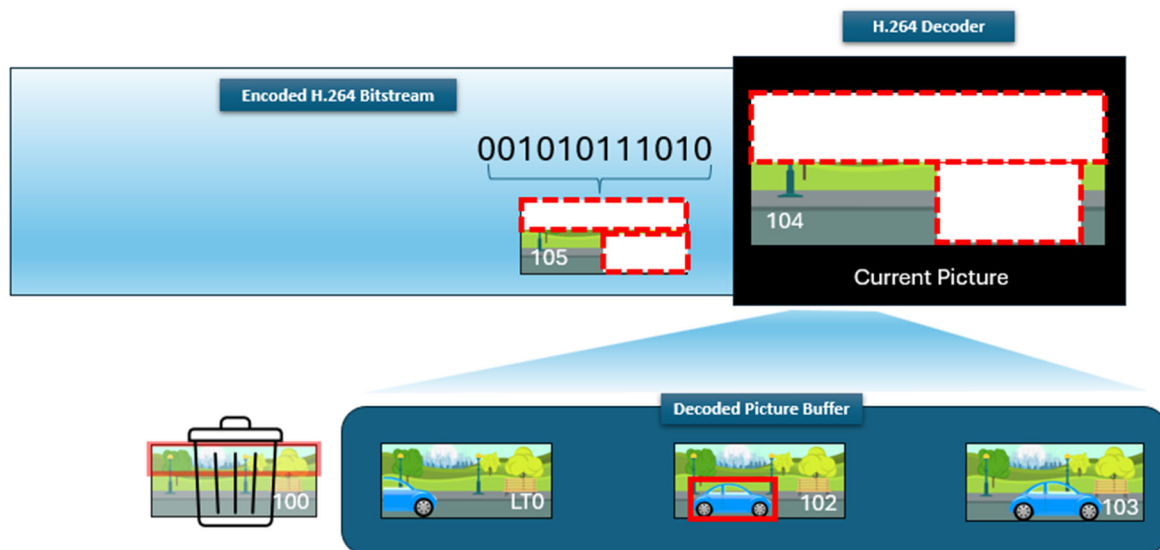
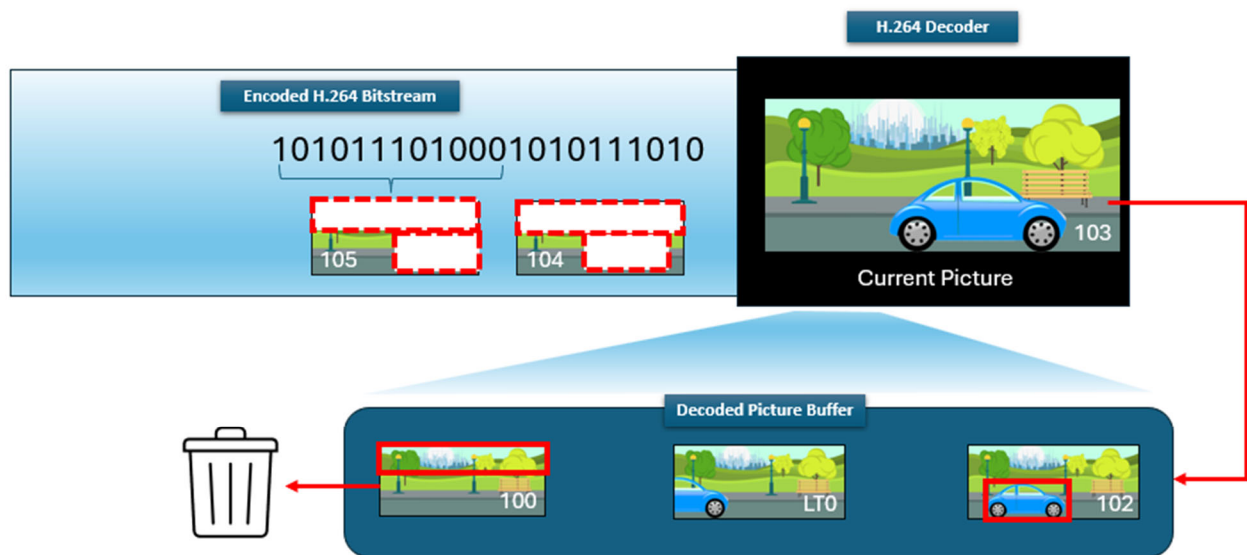
93. As shown above, for example, Picture 100 could be removed from the DPB to make room for Picture 103.

94. In H.264, once a picture is removed from the DPB, it can no longer be used as a reference picture and cannot be added back to the DPB.

95. In H.264, there are two ways in which the decoder determines which pictures to remove from the DPB.

96. *First*, H.264 teaches a sliding window method. The sliding window method is a “first-in, first-out” mechanism whereby the oldest short-term reference picture in the DPB is removed when the DPB is full to make room for the incoming picture. Ex-1005 at § 8.2.5.3. Long-term reference pictures are not removed by the sliding window process.

97. In the following example, the short-term picture that will be removed from the DPB to make room for Picture 103 is Picture 100 because it was the first picture processed.



98. *Second*, H.264 teaches the use of Memory Management Control Operations (MMCO) commands. H.264 defines 7 possible MMCO commands (values 0 through 6) as shown below.

Table 7-9 – Memory management control operation (memory_management_control_operation) values

memory_management_control_operation	Memory Management Control Operation
0	End memory_management_control_operation syntax element loop
1	Mark a short-term reference picture as “unused for reference”
2	Mark a long-term reference picture as “unused for reference”
3	Mark a short-term reference picture as “used for long-term reference” and assign a long-term frame index to it
4	Specify the maximum long-term frame index and mark all long-term reference pictures having long-term frame indices greater than the maximum value as “unused for reference”
5	Mark all reference pictures as “unused for reference” and set the MaxLongTermFrameIdx variable to “no long-term frame indices”
6	Mark the current picture as “used for long-term reference” and assign a long-term frame index to it

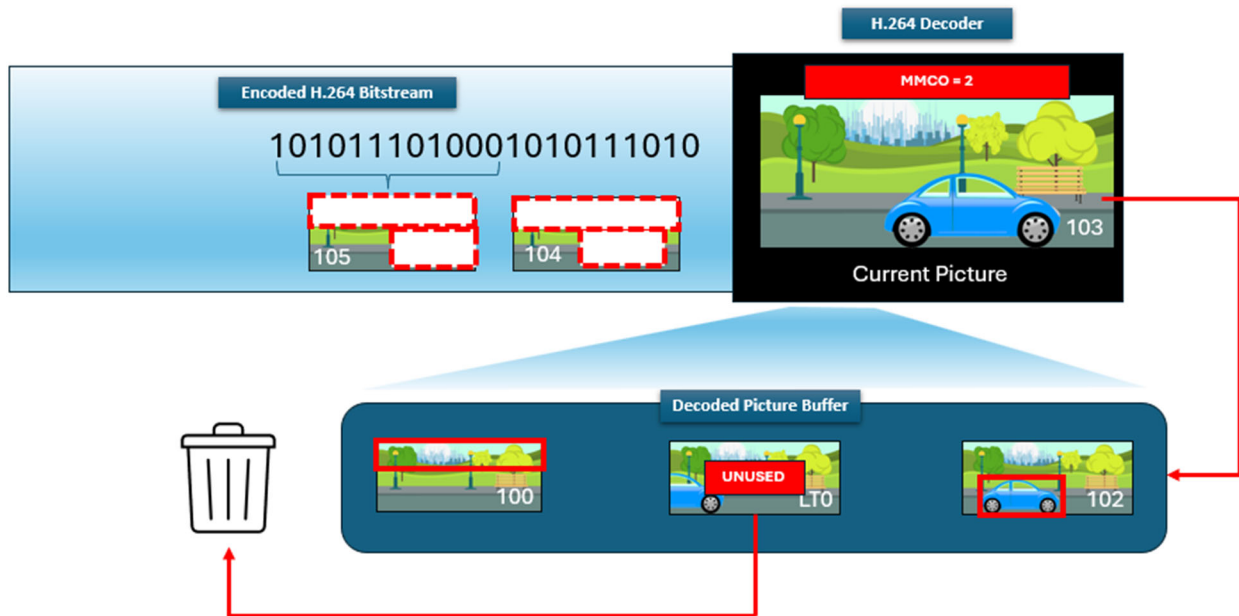
Ex. 1005 at 88.

99. I note that Dr. Creusere suggests that MMCO commands may be used to indicate which pictures should be saved in the DPB. Ex-1003 ¶ 70. This is incorrect. H.264’s MMCO commands cannot choose which pictures should be saved in the DPB, but instead can only remove reference pictures from the DPB (by marking them as “unused for reference”) or change their marking (for example marking a picture as “used for long-term reference” and assigning a long-term frame

index to it). Ex-1005 at 120 (§ 8.2.5.4). This is clear from the excerpt of H.264 above, which provides the complete list of MMCO commands and the types of actions they support.

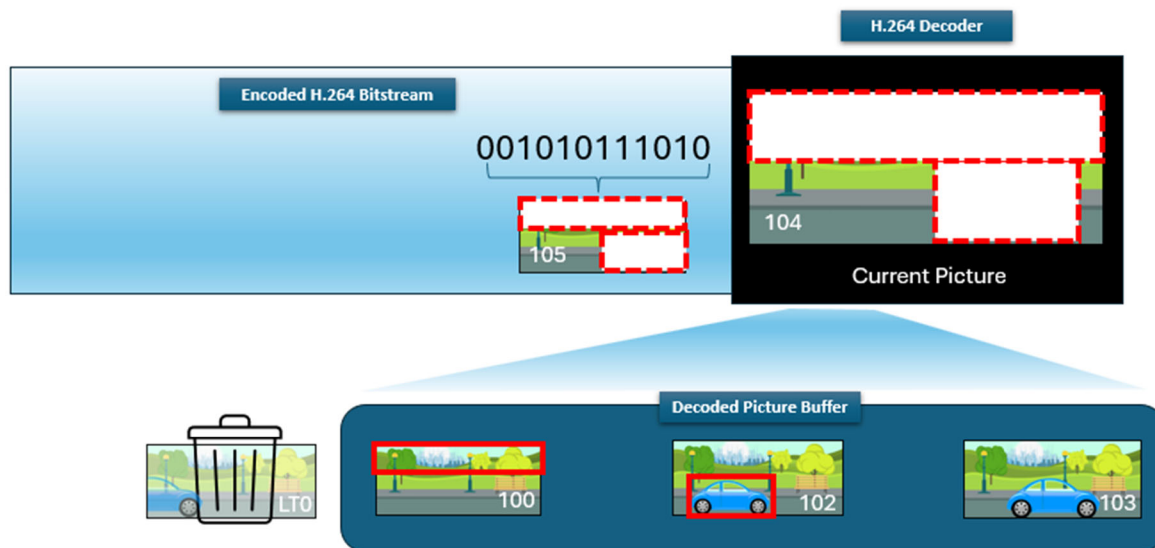
100. MMCO commands in H.264 are sent from the encoder to the decoder as part of the H.264 compressed bitstream. The decoder should perform the corresponding action on reference pictures in its DPB based on the MMCO command it receives. For example, an MMCO command equal to 2 signals that the decoder should mark a long-term reference picture as “unused for reference”, i.e. remove it from the DPB.

101. An example of how MMCO commands could be used is illustrated below:



102. Here, when processing Picture 103, the bitstream also includes an MMCO command of Type 2. Type 2 marks a long-term reference picture as unused for reference. In this example, I have assumed that the long-term reference picture identified in the MMCO is the picture labelled as LT0. When the decoder receives the MMCO command, it should mark Picture LT0 as unused for reference.

103. Thus, when Picture 103 is decoded, as shown below, Picture LT0 is no longer in the DPB. Picture 103 can be stored in the buffer without the decoder having to remove Picture 100.



3. *Common Problems With Using MMCO Commands and Sliding Window in H.264*

104. The use of H.264's sliding window mechanism can lead to problems if the DPB is not maintained correctly. Specifically, absent an MMCO command, the

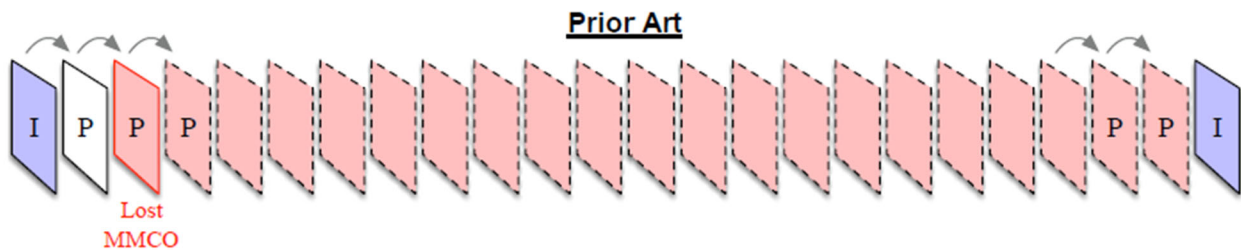
oldest short-term reference picture will be removed regardless of whether it may be needed for a later picture.

105. Thus, one issue with MMCO commands and the sliding window, explained in detail in the '618 Patent (Ex-1001 at cols. 2-3), is that certain MMCO commands (or a picture containing them) can be “lost” during the transmission process, for example due to a transmission error. This can have a very significant impact on subsequent pictures being decoded based on incorrect reference pictures. *See, e.g.*, Ex. 1001 at 2:14-3:16.

106. Considering our earlier example, if the MMCO command that accompanies Picture 103 is lost, then Picture 100 will be deleted from the DPB using the sliding window process, rather than Picture LT0 being deleted as indicated by the missing MMCO command. The result is that the DPB in the encoder will not match the DPB in the decoder, and the entire decoding process could be corrupted until the DPB is reset, for example by the decoding of an IDR picture. Ex-1001 at 2:66-3:2.

107. If the DPB is not reset for a period of many successive pictures, the inconsistency between the encoder and decoder DPB contents can compound significantly. Ex. 1001 at 3:2-6. The illustration below highlights this point. In typical usage of H.264, IDR pictures (i.e. I pictures that do not use reference pictures and which cause the DPB to be reset) are sent at regular intervals, for example once

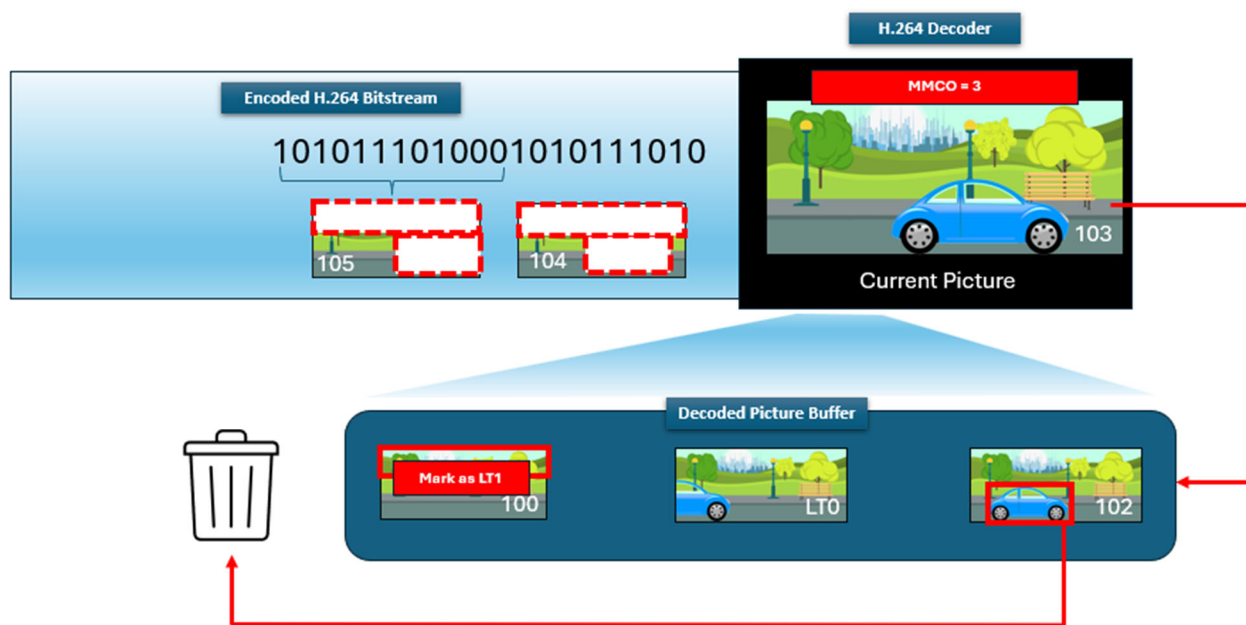
per 25 or more pictures. These IDR pictures are purple in the illustration below. In this example, an MMCO command associated with a P-picture (shown in red) is lost or corrupted due to a transmission error. The decoder's DPB no longer matches the encoder's buffer and subsequent P pictures cannot be correctly decoded. The error cannot be recovered until the next IDR picture. *See* Ex-1001 at 2:56-3:2. In this example, the result will be corrupted or un-decodable P-pictures after the lost MMCO command. These are illustrated as the red dotted boxes below. This in turn is likely to result in a damaged or blank image on the user's screen, perhaps lasting seconds or more.



108. Another issue with the MMCO model pertains especially to long-term pictures. Ex. 1001 at 2:56-64. MMCO commands can be used to remove a short-term or long-term picture from the DPB, but the commands can also be used to change a short-term picture to a long-term picture. This type of MMCO is referred to as Type 3. Ex. 1001 at Table 1.

109. An example of how a Type 3 MMCO command works and the potential issues with losing this type of command is as follows. In this example, an MMCO

command type 3 is sent and instructs the decoder to change short-term (ST) picture having PicNum 100 to a long-term (LT) picture with LongTermFrameIndex 1 (the next available LongTermFrameIndex).



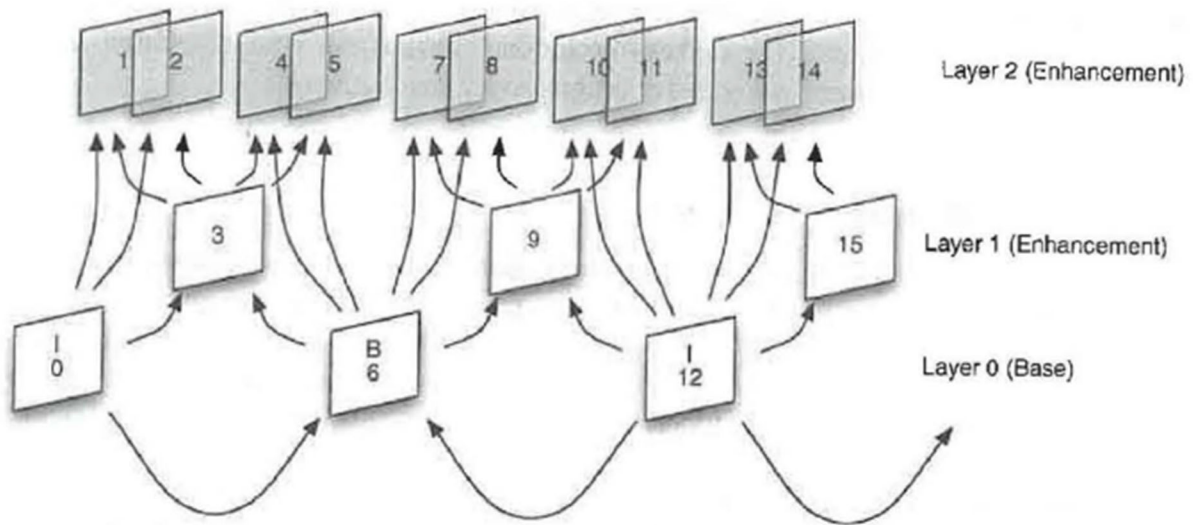
110. Because what was short-term Picture 100 is now a long-term picture (LT1), the sliding window mechanism does not apply to it. Instead, the next short-term picture (Picture 102) is removed from the DPB to make room for Picture 103.

111. The decoder needs to know that LT1 (the picture with LongTermFrameIndex 1) corresponds to the picture data that originally had a PicNum of 100, i.e. was previously indexed as 100. Thus, LT1 is a relative index that depends on a correctly received MMCO, in this case the MMCO command that changed Short Term picture 100 to Long Term picture LT1. A long-term reference

frame such as LT1 can only be successfully used for prediction if the decoder knows which picture data it corresponds to, i.e. if the MMCO command that marked the frame as “used for long term reference” was correctly received by the decoder.

112. If this Type 3 MMCO command is lost, then the decoder never changes the short-term picture (Picture 100) to a long-term picture (LT1), i.e. it never marks Picture 100 as “used for long term reference” with long-term frame index LT1. In that case, Picture 100 would be removed from the DPB by the sliding window method instead of Picture 102 because Picture 100 would never have been marked as a long-term picture and would continue to be treated as a short-term picture. The decoder cannot thereafter correctly predict any blocks or portions of any picture that uses prediction from LT1 (formerly Picture 100).

113. As explained in the '618 Patent, lost MMCO commands present a particular problem when using temporal scalability. Ex. 1001 at 13:44-14:20. Temporal scalability, which is supported by H.264, allows the use of multiple temporal layers of pictures in a bitstream. Ex. 2003 at 290-293. This allows flexibility from the encoder and decoder to opt not to decode certain layers. For example, a bitstream could be designed to have a base layer (Layer 0) and one or more enhancement layers. An example is shown below.



Ex. 2003 at 293.

114. Using temporal scalability, the base layer may be coded at a low temporal resolution or a low frame rate, and enhancement layers may be added to increase the frame rate of the decoded sequence. Ex-2003 at 290. This hierarchical structure is illustrated above.

115. A benefit of temporal scalability is that a decoder can choose to only decode the base layer or otherwise fewer than all the layers, which requires fewer bits to decode the video and is particularly useful when there is limited bandwidth and/or if the decoding device cannot decode the full frame rate in real time. The more layers that are decoded, the better the quality of the video, but the less compression savings are obtained and the greater the computational burden on the decoding device.

116. In H.264/AVC, one issue with the relative signaling model for handling the DPB is that it can significantly impact temporal scalability. The inventors explained this problem in the '618 Patent. *See* Ex. 1001 at 13:44-14:20. If a temporal layer is removed (for example by a network mode that is designed to only decode certain temporal layers), then there will be gaps in the frame number indicated by the syntax element `frame_num`. *Id.* In H.264/AVC, not only would all those pictures be discarded, but so would their MMCO commands along with them, potentially affecting the decoding of the other layers. Thus, as the '618 Patent explains, “prior art solutions have problems with temporal scalability for certain coding structures since the information about the reference pictures for the current picture is affected by the removal of a previous picture from the bitstream as the reference picture information is signaled in a relative way.” Ex-1001 at 14:15-20.

4. *Reference Picture Lists*

117. In addition to the use of a DPB, H.264 supports the use of reference picture lists in the decoding process. In H.264, a reference picture list is a conceptual list that contains reference picture indices pointing to all the pictures in the DPB in a specific order. Note that a reference picture list does not change the contents of the DPB itself, including the decoded picture data and associated indicators such as ST100 or LT1.

118. For every picture, before that picture is decoded, the H.264 decoder indexes all the decoded pictures in the DPB into reference picture list(s). If the current picture is a P-picture, one reference picture list is created, and if the current picture is a B-picture, two reference picture lists are created. Ex. 2003 at 107. No reference picture list is generated for an I-picture because it does not use any reference pictures.

119. The pictures in the reference picture lists are indexed starting with the short-term pictures followed by any long-term pictures. The short-term pictures are indexed, according to this set of default rules, where POC is Picture Order Count and indicates the display order of decoded pictures:

List0 (P slice) : The default order is in decreasing order of *PicNum*. *PicNum* is a version of *frame_num*, 'wrapped around' modulo *MaxFrameNum*.

List0 (B slice) : The default order is (1) in decreasing order of POC, for pictures with POC earlier than current picture, then (2) in increasing order of POC, for pictures with POC later than the current picture.

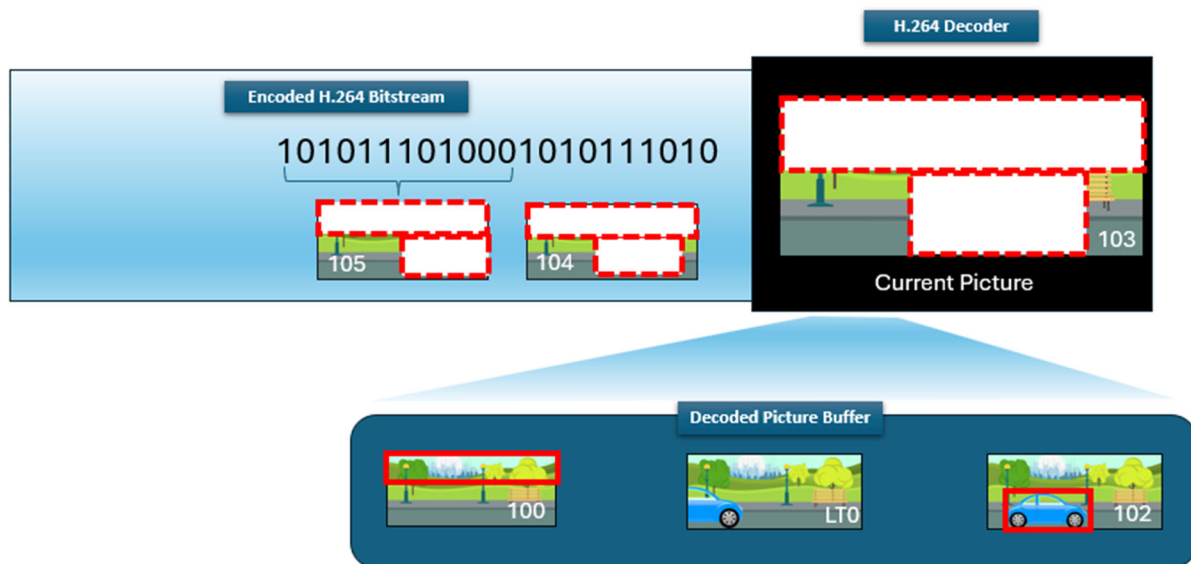
List1 (B slice) : The default order is (1) in increasing order of POC, for pictures with POC later than the current picture, then (2) in decreasing order of POC, for pictures with POC earlier than current picture.

Ex-2003 at 107.

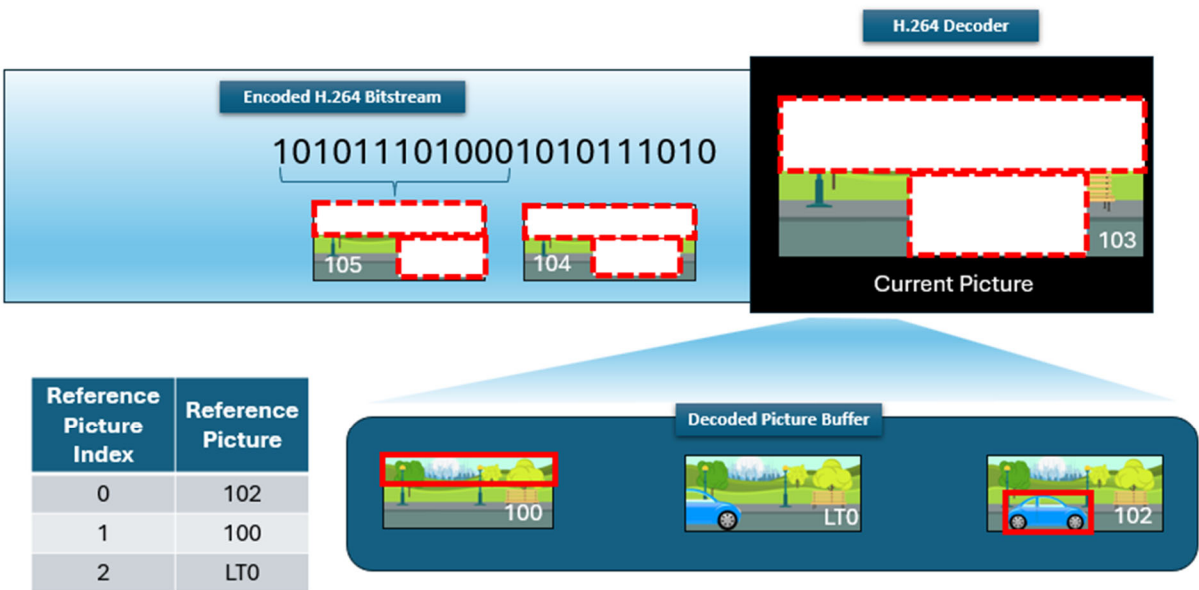
120. The long-term pictures are indexed in the reference picture list(s) in ascending order of LongTermPicNum. Thus the default reference picture list index order for a P slice depends on decoding order and the default order for a B slice depends on display order.

121. These default rules are designed so that the frames most likely to be used as reference pictures for the current frame are nearer the start of the list. The order of the lists is important because it costs more bits to signal a reference picture that is lower down the reference picture list. Ex-2003 at 107.

122. Consider our example. Before decoding Picture 103, the decoder must construct Picture 103's reference picture list. The below illustration shows the state of the decoder before that list is constructed:



123. After the reference picture list is constructed, the state of the decoder looks like this:



124. I provide a further example of reference picture list organisation in in my 2010 book on H.264, which I also include here for reference:

Reference Picture Order Example: P slices (list0)

The reference picture list is initially empty. The current *frame_num* is 150 and the maximum size of the DPB is 5 frames. Italics indicate a *LongTermPicNum*. Notice that by default, the first entry in list0 is the most recently coded frame. This is likely to be the ‘best’ prediction reference for the majority of macroblocks.

Operation	list0(0)	list0(1)	list0(2)	list0(3)	list0(4)
Initial state	-	-	-	-	-
Encode frame 150	150	-	-	-	-
Encode 151	151	150	-	-	-
Encode 152	152	151	150	-	-
Encode 153	153	152	151	150	-
Encode 154	154	153	152	151	150
Encode 155	155	154	153	152	151
Assign 154 to LongTermPicNum 3	155	153	152	151	3
Encode 156 and mark it as LongTermPicNum 1	155	153	152	1	3
Encode 157	157	155	153	1	3

Ex. 2003 at 107.

125. If using the same table format used in my example above, the reference picture list would look like this for the sixth picture in the bitstream (once all five pictures in this example have been decoded). Note that “LT” stands for “long-term picture” and corresponds with the italicized index numbers in the table above. “ST” stands for “short-term picture.”

Reference picture index	Reference picture
0	ST157
1	ST155
2	ST153
3	LT1
4	LT3

126. Note that frame_num may be a relatively large value (150-157 in this example) and LongTermPicNum is typically a small value (1 or 3 in this example). This is because LongTermPicNum is not an absolute value—it is an index value. The first picture to be characterized as a long-term picture may be given the index LT0, the next picture may be given the index LT1, the next given the index LT2, etc.

5. *Reference Picture List Reordering Commands*

127. The default order for the initial reference picture list may not be the most efficient order for encoding a particular frame, for example if an older reference picture in the DPB happens to be a particularly good prediction source for the current frame. To handle this situation, H.264 supports “reference picture list reordering commands.” Ex. 1005 at 47 (§7.3.3.1), 85-86 (§ 7.4.3.1).

128. Reference picture list reordering may be used to modify the default order of each reference picture list. H.264’s reference picture list reordering commands modify the order of the *reference picture lists* and do not change the contents of the DPB. As I explained in my 2010 book, “reference pictures are ordered in one or two lists prior to encoding or decoding a slice” and “the default order... may be modified by explicit commands in the slice header”. Ex 2003 at 107, 109.

129. For example, H.264 states that the reference picture list reordering instructions in H.264 “allow[] the contents of RefPicList0 and for B slices RefPicList1 to be modified in a flexible fashion.” Ex. 1005 at 112 (§ 8.2.4).

130. The reference picture list reordering syntax is reproduced below.

7.3.3.1 Reference picture list reordering syntax		
ref_pic_list_reordering() {	C	Descriptor
if(slice_type != I && slice_type != SI) {		
ref_pic_list_reordering_flag_l0	2	u(1)
if(ref_pic_list_reordering_flag_l0)		
do {		
reordering_of_pic_nums_idc	2	ue(v)
if(reordering_of_pic_nums_idc == 0		
reordering_of_pic_nums_idc == 1)		
abs_diff_pic_num_minus1	2	ue(v)
else if(reordering_of_pic_nums_idc == 2)		
long_term_pic_num	2	ue(v)
} while(reordering_of_pic_nums_idc != 3)		
}		
if(slice_type == B) {		
ref_pic_list_reordering_flag_l1	2	u(1)
if(ref_pic_list_reordering_flag_l1)		
do {		
reordering_of_pic_nums_idc	2	ue(v)
if(reordering_of_pic_nums_idc == 0		
reordering_of_pic_nums_idc == 1)		
abs_diff_pic_num_minus1	2	ue(v)
else if(reordering_of_pic_nums_idc == 2)		
long_term_pic_num	2	ue(v)
} while(reordering_of_pic_nums_idc != 3)		
}		
}		
}		

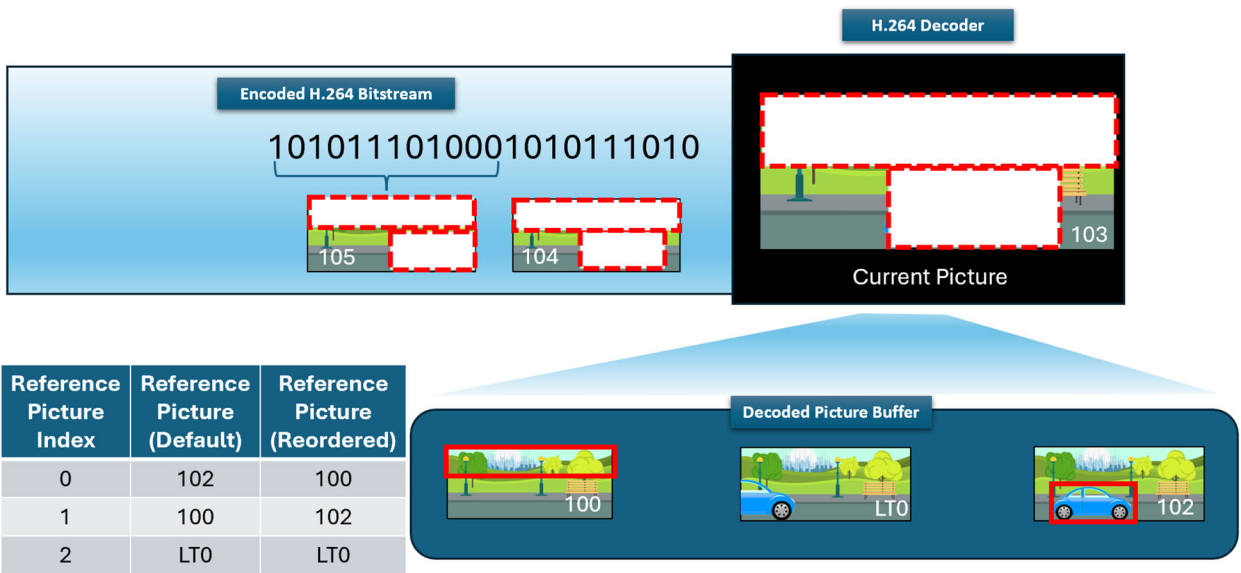
Ex. 1005 at 47.

131. This reordering section refers to the “abs_diff_pic_num_minus1” syntax element. The H.264 standard provides that the syntax element “abs_diff_pic_num_minus1 plus 1 specifies the absolute difference between the picture number of the picture being moved to the current index in the list and the picture number prediction value.” Ex. 1005 at 85-86 (§ 7.4.3.1). In H.264, the value abs_diff_pic_num_minus1 plus 1 is added to or subtracted from the picture number prediction value to identify a reference picture to be moved to a current index in the reference picture list.

132. The H.264 standard defines three of the reordering command syntax elements (reordering_of_pic_nums_idc, abs_diff_pic_num_minus1, and long_term_pic_num) as specifying “the change from the initial reference picture lists to the reference picture lists to be used for decoding the slice.” Ex. 1005 at 96.

133. Thus, in H.264, the reference picture list reordering syntax represents the change from the default order of the initial reference picture list to the new order of the reference picture list. Because the default order of the reference picture list is simply a reordering of whatever available pictures were in the DPB, the default order of the reference picture list is dependent on the DPB being correct, i.e. on the decoder DPB matching the encoder DPB. In this way, the reference picture list reordering of H.264 is relative.

134. Using our example, the encoder could send reordering commands that signal Pictures 100 and 102 should swap places in the reference picture list:



135. Note that the reference picture list reordering commands do not change the contents of the DPB at all. They only change the order of indexing the reference pictures in the reference picture list.

136. I provide a further example of the use of reference picture reordering commands in my 2010 book, which I have reproduced below. Ex. 2003 at 109-10.

5.3.3.2 Changing the reference picture list order

The default order of list0, and of list1 in the case of a B slice, may be modified by explicit commands in the slice header. These commands are initiated by the syntax element *ref_pic_list_reordering_flag* and change the list order **for the current slice only**. This may be used, for example, to enable an encoder to place a particularly useful reference frame earlier in the list than its usual position, since earlier reference indices cost fewer bits to signal.

The reference picture list re-ordering process is illustrated in Figure 5.8 and proceeds as follows for list 0. This is a simplified overview.

- Initialize a pointer (*refIdxL0*) to point to the first reference picture index in the list (0)
- While *reordering_of_pic_nums_idc* $\neq$ 3
 - o Select a reference picture, either short term, indicated by *abs_diff_pic_num_minus1*, or long term, indicated by *long_term_pic_num*
 - o Move all other pictures from position *refIdxL0* onwards in the list to one position later
 - o Put this picture in the list at the position indicated by *refIdxL0*
 - o Increment pointer *refIdxL0*.

Table 5.3 Selecting a reference picture to re-map

<i>reordering_of_pic_nums_idc</i>	remappedPicNum is calculated as:
0	$\text{predictedPicNum} - (\text{abs_diff_pic_num_minus_1} + 1)$
1	$\text{predictedPicNum} + (\text{abs_diff_pic_num_minus_1} + 1)$
2	<i>long_term_pic_num</i>
3	(Exit the re-ordering process)

The value *abs_diff_pic_num_minus1*+1 signals a positive or negative offset from a **predicted** reference picture number. For the first reordering or remapping instruction, the predicted picture number *predictedPicNum* is the current picture number. For subsequent reordering instructions, the predicted picture number is the picture number of the last remapped picture. The picture to be remapped or moved, *remappedPicNum*, is selected according to the following rules shown in Table 5.3.

Reference Picture Re-Ordering Example:

The current slice is a P slice. The DPB contains five reference frames, indexed in list0. Three of these are short-term frames with *PicNum* equal to 153, 155, and 157, respectively. The other two are long-term frames with *LongTermPicNum* equal to 1 and 3, respectively. The current *frame_num* is 158. The default reference list order (list0) is as follows (Table 5.4).

Table 5.4 Initial reference picture list

Index	0	1	2	3	4
Picture	157	155	153	1	3

The following series of four reference picture reordering commands are received.
Initial *predictedPicNum* = 158, the current picture; initial *refIdxL0* = 0

1. *reordering_of_pic_nums_idc* = 0, *abs_diff_pic_num_minus_1* = 4
 $\text{remappedPicNum} = 158 - 5 = 153$
Put picture 153 at position *refIdxL0*=0 (the start of the list).
New list: **153**, 157, 155, 1, 3
New *predictedPicNum* = 153; new *refIdxL0* = 1.
2. *reordering_of_pic_nums_idc* = 1, *abs_diff_pic_num_minus_1* = 1
 $\text{remappedPicNum} = 153 + 2 = 155$
Put picture 155 at position *refIdxL0*=1, second in the list.
New list: 153, **155**, 157, 1, 3
New *predictedPicNum* = 155; new *refIdxL0* = 2.
3. *reordering_of_pic_nums_idc* = 2, *long_term_pic_num* = 3
 $\text{remappedPicNum} = 3$
Put long term picture 3 at position *refIdxL0*=2, third in the list.
New list: 153, 155, **3**, 157, 1
4. *reordering_of_pic_nums_idc* = 3
End of reordering process.

The final list0 order is as follows (Table 5.5).

Table 5.5 Final reference picture list

Index	0	1	2	3	4
Picture	153	155	3	157	1

137. Or to use the format of my earlier examples, the default reference picture list and the reordered reference picture list are as follows, with short-term reference picture numbers denoted as ST_ and long-term reference picture numbers denoted as LT_:

Reference picture index	Reference picture (default)	Reference picture (reordered)
0	ST157	ST153
1	ST155	ST155
2	ST153	LT3
3	LT1	ST157
4	LT3	LT1

138. Notice that the default reference picture list order for a P-picture **always** comprises first the short-term reference pictures in descending order of PicNum (which equals frame_num if the video sequence is progressive and not interlaced), followed by the long-term reference pictures in ascending order of LongTermPicNum.

139. I note that Dr. Creusere provides a purported example of H.264 operation (Ex. 1003 at ¶¶ 117-24) in which an initial or default reference picture list order is (ST149, ST138, ST144, LT12, LT17). In my view this is a misleading example because H.264 cannot support short-term reference pictures in a P picture in such a default reference picture list order. I address this in more detail below in Section XII.D.

X. U.S. PATENT NO. 10,708,618 (“’618 Patent”) (Ex-1001)

A. Priority Date

140. The ’618 Patent claims priority to U.S. provisional application No. 61/503,019, which has a filing date of June 30, 2011.

141. Petitioner does not challenge this priority date for purposes of the Petition, and I have assumed that each of the challenged claims is entitled to a priority date of June 30, 2011. Pet. at 11.

B. '618 Patent Overview

142. United States Patent No. 10,708,618 discloses a novel way of absolute reference picture signaling that was adopted into the H.265 Standard, as I discuss later in Paragraphs 192-203.

143. The '618 Patent begins by describing the reference picture mechanisms in the H.264 standard and their problems. The patent explains that H.264 has two processes to signal which pictures are to be stored in the decoded picture buffer. Ex. 1001 at 2:3-13. In one of the processes, “the slice headers of the picture contain Memory Management Control Operations (MMCO) commands.” *Id.* In the MMCO model, “adaptive memory control process is applied after decoding of the picture to obtain relative reference to the pictures to be stored in the decoded picture buffer.” *Id.* Otherwise, the second process (the sliding window process) is applied to obtain relative reference to the pictures to be stored in the decoded picture buffer. *Id.* The MMCO commands supported by H.264 are listed in the background section of the '618 Patent.

TABLE 1

Memory management control operation values for H.264/AVC	
memory_management_control_operation	Memory Management Control Operation
0	End memory_management_control_operation syntax element loop
1	Mark a short-term reference picture as “unused for reference”
2	Mark a long-term reference picture as “unused for reference”
3	Mark a short-term reference picture as “used for long-term reference” and assign a long- term frame index to it
4	Specify the maximum long-term frame index and mark all long-term reference pictures having long-term frame indices greater than the maximum value as “unused for reference”
5	Mark all reference pictures as “unused for reference” and set the MaxLongTermFrameIdx variable to “no long-term frame indices”
6	Mark the current picture as “used for long- term reference” and assign a long-term frame index to it

Ex-1001 at Table 1.

144. The '618 Patent explains that “[a] problem with H.264/AVC is its vulnerability to losses of pictures.” *Id.* at 2:14-16. According to the '618 Patent, “if a picture containing MMCO of type 2, 3, 4, 5, or 6 is lost there is a risk that the number of long term pictures in the DPB is different from what it would have been if the picture was received, resulting in an ‘incorrect’ sliding window process for all the following pictures.” *Id.* at 2:56-60. In other words, “the encoder and decoder will contain a different number of short-term pictures resulting in out-of-sync behavior of the sliding window process.” *Id.* at 2:60-63.

145. According to the '618 Patent, “[w]hat makes the situation even worse is that a decoder will not necessarily know that the sliding window process is out-of-sync and thus cannot report the problem to the encoder or request an IDR [i.e., I] picture even in applications where a feedback channel is available.” *Id.* at 3:2-6.

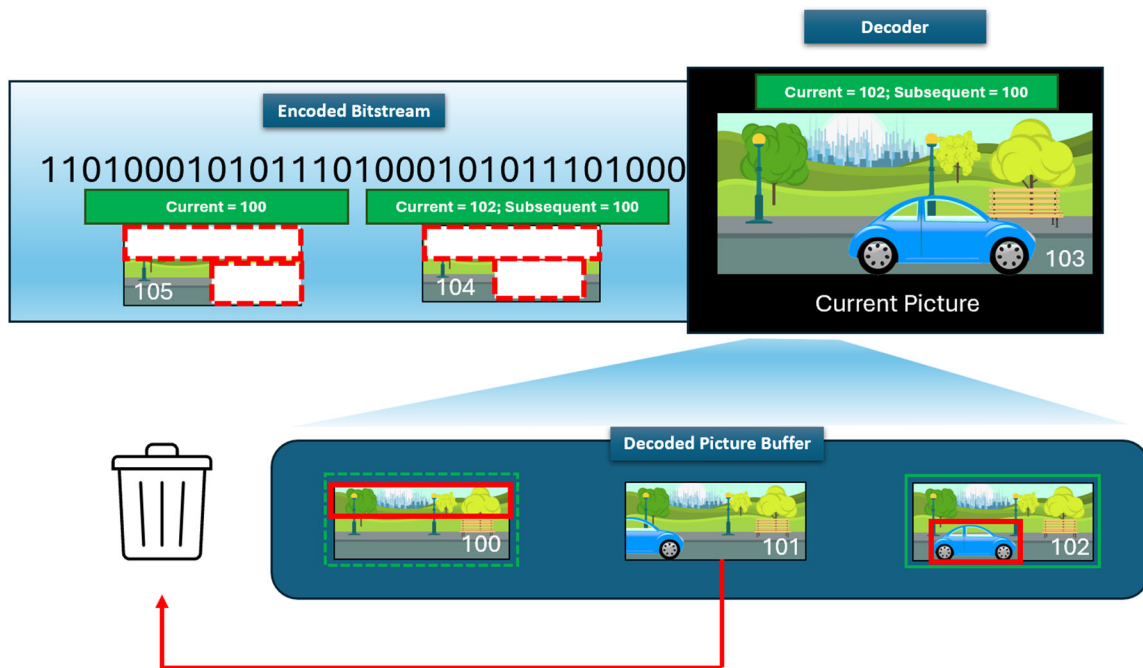
146. Thus, the '618 Patent explains that the H.264 Standard uses a relative reference picture signaling mechanism that depends on previously decoded pictures having been decoded accurately and that depends on the previous state of the DPB.

147. Given the problems associated with the relative reference picture signaling techniques in H.264, the '618 Patent identifies “a need for an efficient reference picture signaling and buffer management that do not suffer from the shortcomings and limitations of prior art solutions.” *Id.* at 3:14-16.

148. The '618 Patent provides that solution: “[i]n clear contrast to the prior art solutions in which correct reference picture management is dependent on that previously encoded pictures have been correctly received and decoded, the embodiments provide buffer description information that is used for reference pictures in an absolute and explicit way instead of a relative or implicit way.” *Id.* at 4:44. In this way, “the encoded representation of a picture contains the information about which reference pictures to use for reference during decoding independent of the encoded representations of previous pictures in the video stream.” *Id.* at 4:44-48.

149. The '618 Patent explains that this “buffer description information,” or “absolute and explicit way” of reference picture signaling, is “also referred to as Reference Picture Set (RPS).” *Id.* at 8:22-25. This buffer description information identifies exactly which pictures are to be used for current pictures and which are to be used for subsequent pictures. *Id.* at 7:6-10. The patent explains that the picture identifiers in the buffer description information must “unambiguously identify the picture,” for instance identifying display order (e.g. Picture Order Count, POC), decoding order (e.g. frame number, frame_num) or another identifier that is not relative. *Id.* at 7:48-53. As the patent explains, “the buffer description identified or obtained by the buffer description information enables an unambiguously[sic] identification of the relevant reference picture(s).” *Id.* at 7:62-64.

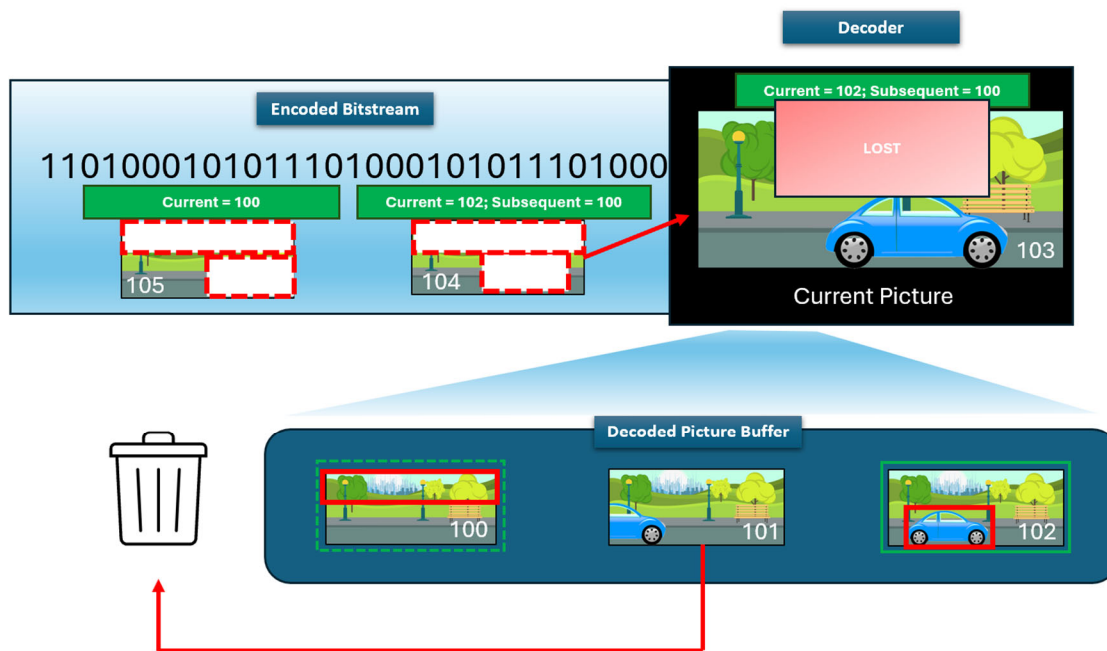
150. The illustration below shows how the buffer description information in the '618 Patent operates in place of MMCO commands and the sliding window.



151. A key difference between this illustration (illustrating the buffer description information in the '618 Patent) and the DPB management disclosed in the H.264 Standard is that each picture in the '618 Patent will include buffer description information or an RPS (illustrated by a solid green box above), which unambiguously identifies any reference pictures that will be used to decode the current picture (e.g. picture 102) and any reference pictures that will be used to decode a subsequent picture (e.g. picture 100). Another key difference is that the pictures are all identified by their picture order count (POC) or other unambiguous identifier values, regardless of whether they are a short-term picture (Pictures 100 and 102) or long-term picture (Picture 101). Ex-1001 at 8:1-9.

152. One benefit of using buffer description information, or absolute and explicit reference picture signaling, instead of MMCO commands or sliding window, is that the process using buffer description information is not as vulnerable to loss. Ex-1001 at 4:49-54. Because the buffer description information is communicated directly in the slice header, i.e. in the encoded representation of the picture, it does not depend on any previously received information. This is in contrast with H.264, where for example the correct mapping of a long-term picture index such as LT1 to a decoded picture with a particular PicNum such as PicNum 100, may depend on a previously received MMCO.

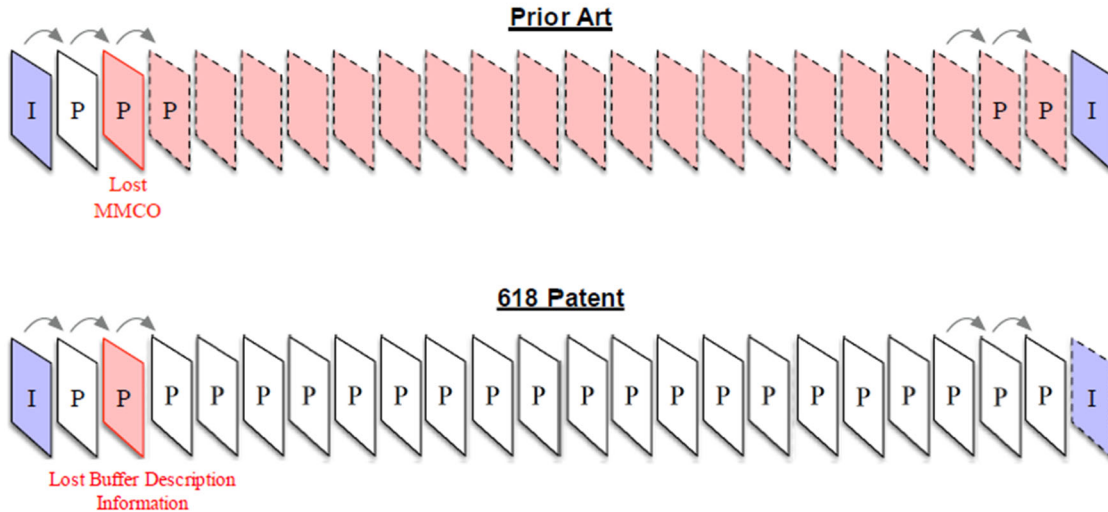
153. For example, using the approach of the '618 Patent, if the buffer description information sent with Picture 103 is lost, the DPB is simply updated at the next picture. There will be no automatic removal of Picture 100 with the sliding window. The next picture will simply have its own buffer description information listing all the reference pictures necessary for those current and subsequent pictures.



154. Thus, if Picture 103 in the above illustration is lost, Picture 104's buffer description information will provide the correct DPB information for that picture. Picture 100 will not be incorrectly removed from the DPB by a default sliding window process.

155. The effect of a lost picture on the DPB when using the '618 Patent's invention does not propagate like it would in systems using relative signaling mechanisms, such as H.264. This difference is illustrated below. If buffer description information communicated with Picture 102 is lost, the correct buffer description information is received again with Picture 103. Thus, if part or all of picture 103 is lost, the decoder can most likely (i) detect a problem with the DPB at Picture 103 and/or (ii) receive correct DPB information with Picture 104 such that all future

pictures can be successfully decoded (albeit perhaps with error concealment for any blocks that happen to use Picture 103 as a reference picture).



156. The '618 Patent also explains that its absolute and explicit signaling solves the problem with temporal scalability that I discussed earlier in Paragraphs 113-16. The patent states that “prior art solutions have problems with temporal scalability for certain coding structures since the information about the reference pictures for the current picture is affected by the removal of a previous picture from the bitstream as the reference picture information is signaled in a relative way.” Ex-1001 at 14:15-20. The embodiments of the '618 Patent solve this temporal scalability problem because “[u]sing the embodiments, temporal scalability will not be restricted as in the prior [art] since the reference picture for the current picture is not affected by removal of previous pictures from the bitstream since it is signaled in an absolute way.” *Id.* at 14:20-24.

157. Another benefit of the '618 Patent explicitly identifying which pictures are used as reference pictures for the current picture and which pictures are used for reference for a subsequent picture is that the decoder need only construct the reference picture list with the reference pictures necessary for the current picture—not all the pictures like in the prior art method. This can save bits leading to improved compression efficiency. For example, if the picture with PicNum 106 is decoded and marked as a reference picture for a subsequent picture only, it need not be included in the reference picture list for the current picture, i.e. it is not assigned a reference picture index for the decoding of the current picture. This can reduce the number of bits required to signal reference picture indices, which may be a significant saving as such indices may need to be signaled many times during the encoding of an inter-coded picture.

158. The '618 Patent provides example embodiments that incorporate its inventive concepts. For example, one embodiment relates to a method comprising “retrieving buffer description information defining at least one reference picture from the encoded representation of the picture.” *Id.* at 3:27-30. Further, “[a]t least one picture identifier identifying a respective reference picture of the at least one reference picture is determined based on the buffer description information.” *Id.* at 3:30-32. “The respective reference picture is to be used as decoding reference for

the picture,” and “[a] decoded picture buffer is updated based on the determined at least one picture identifier.” *Id.* at 3:32-35.

159. “A related aspect of the embodiments defines a decoder configured to decode an encoded representation of a picture of a video stream of multiple pictures.” *Id.* at 3:36-38. This “decoder comprises a data retriever configured to retrieve buffer description information defining at least one reference picture from the encoded representation of the picture,” and “a picture identifier determiner of the decoder is configured to determine, based on the buffer description information, at least one picture identifier identifying a respective reference picture of the at least one reference picture as decoding reference for the picture.” *Id.* at 3:38-46. This “decoder also comprises a buffer manager configured to update a decoded picture buffer based on the at least one picture identifier.” *Id.* at 3:46-48.

160. The ’618 Patent explains that these “embodiments do not have these problems of the prior art techniques by using a fundamentally different approach for signaling reference pictures as compared to the prior art.” *Id.* at 6:38-41. According to the ’618 Patent, these “embodiments instead specify which decoded pictures to be used for reference pictures in an absolute or explicit way instead of a relative or implicit way.” *Id.* at 6:41-43. To do so, “the encoded representation, i.e. the bitstream, for a current picture contains the information about what pictures to use

for reference, i.e. reference pictures, independent of the encoded representations of previous pictures.” *Id.* at 6:44-48.

161. The ’618 Patent further explains that “[a]ccording to an aspect of the embodiments a method is provided that creates buffer description information, e.g. a data structure, such as a general table with absolute information of which pictures to be used for reference pictures, i.e. stored in a decoded picture buffer (also referred to as a reference picture buffer) to be used for subsequent coding.” *Id.* at 6:57-62. It states that “the buffer description identified or obtained by the buffer description information enables an unambiguously [sic] identification of the relevant reference picture(s).” *Id.* at 7:62-64.

162. The ’618 Patent describes how the encoder inserts the buffer description information into the bitstream. First, “at least one reference picture of a video stream of multiple pictures is determined as encoding reference.” *Id.* at 6:65-7:4. Next, the encoder provides “a respective picture identifier for each reference picture of the at least one reference picture determined in step S1.” *Id.* at 7:38-40. The ’618 Patent explains that “[t]he picture identifier is used, possibly together with other data, to unambiguously identify a reference picture.” *Id.* at 7:40-42. In this way, “the picture identifier can be regarded as an absolute reference to pictures to be used as reference pictures,” meaning “that it is possible to correctly identify the relevant reference picture given its picture identifier and optionally the other data.”

Id. at 7:42-46. The patent states that “[e]xamples of such picture identifiers include Picture Order Count (POC), frame number (frame_num) or POC and an additional identifier (additional_picture_id).” *Id.* at 7:54-56. These are all unambiguous, or absolute identifiers, that do not depend on previously decoded information. This is in contrast to H.264’s long-term picture index values, which are relative identifiers as described in Paragraphs 109-12 and 152.

163. The next step is that the encoder “generates information of a buffer description, also referred to as Reference Picture Set (RPS),” or also “denoted buffer description information.” *Id.* at 8:22-31. According to the ’618 Patent, “[t]he buffer description information is generated based on the picture identifier(s) provided in step S2,” and it “defines, preferably unambiguously defines, the at least one reference picture determined in step S1.” *Id.* at 8:25-29. Because of this, “it is possible to derive the respective picture identifier of the at least one reference picture from the buffer description information.” *Id.* at 29-31.

164. The fourth step the encoder performs is to insert the “generated buffer description information” in an encoded representation of the current picture.” *Id.* at 8:32-38. Thus, “the encoded picture therefore carries the buffer description information that can be used at the decoder to define and identify the reference pictures that are need to decode the current picture and/or any subsequent picture of the video stream.” *Id.*

165. In one embodiment, “the buffer description information [] comprises a listing of the picture identifier(s) provided in step S2 or data allowing calculation of the picture identifier(s) provided in step S2.” *Id.* at 9:21-24. As explained above, these identifiers are unambiguous, for example identifying picture data in the DPB using the display count or decoding order of each identified picture. Ex-1001 at 7:40-46. An alternative embodiment, which “is generally more bit efficient, i.e. generally requires fewer number of bits or symbols for defining the picture identifiers, is to signal the reference picture properties, i.e. picture identifiers, relative to the value of these properties as signaled for the current picture.” *Id.* at 9:32-36. In other words, “a difference is calculated for each picture identifier provided in step S2, between the picture identifier and a picture identifier identifying the current picture.” *Id.* at 9:44-47. The result of this is a “delta identifier.” *Id.* at 9:47-49.

166. The '618 Patent explains that the buffer description information can be “included in the slice header 65, another picture header or another data structure specified by the standard to which the encoder and decoder conforms.” *Id.* at 10:22-25.

167. In addition to describing the encoding process, the '618 Patent also describes the decoding process. For example, it describes a method “in which an encoded video or data stream is received, buffer description information is detected in the encoded video stream and absolute reference to pictures to be used as reference

pictures for decoding a current picture or as reference pictures for future decoding is determined based on the buffer description information.” *Id.* at 15:5-11. Then, “[t]he pictures to be used as reference pictures are stored in a decoded picture buffer.” *Id.* at 15:11-13.

168. In the decoding process, the first step is “where buffer description information defining at least one reference picture is retrieved from the encoded representation of the picture.” *Id.* at 15:15-18. Next, “[t]he buffer description is used in step S31 to determine at least one picture identifier that, preferably unambiguously, identifies a respective reference picture as decoding reference for the picture.” *Id.* at 15:18-22. Next, “[t]he at least one reference picture identifier determined in step S31 is used in step S32 to update a decoded picture buffer.” *Id.* at 15:22-24.

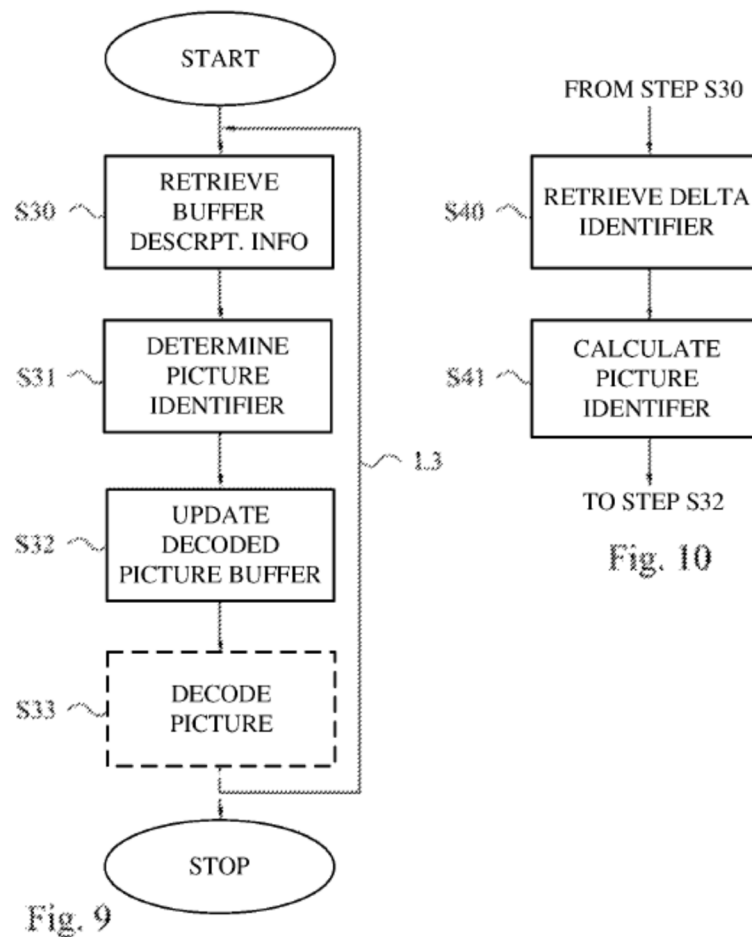
169. After the decoded picture buffer is updated, the next step is decode the picture “based on the encoded representation of the picture and at least one of the reference pictures comprised, present or stored in the decoded picture buffer and identified based on the buffer description information.” *Id.* at 15:25-31.

170. The ’618 Patent discloses that in one embodiment, “[t]he buffer description information could explicitly include the picture identifier(s) of the reference picture(s).” *Id.* at 16:11-12. But “[i]n an alternative embodiment, the buffer description information comprises delta values or delta identifiers.” *Id.* at 16:15-16.

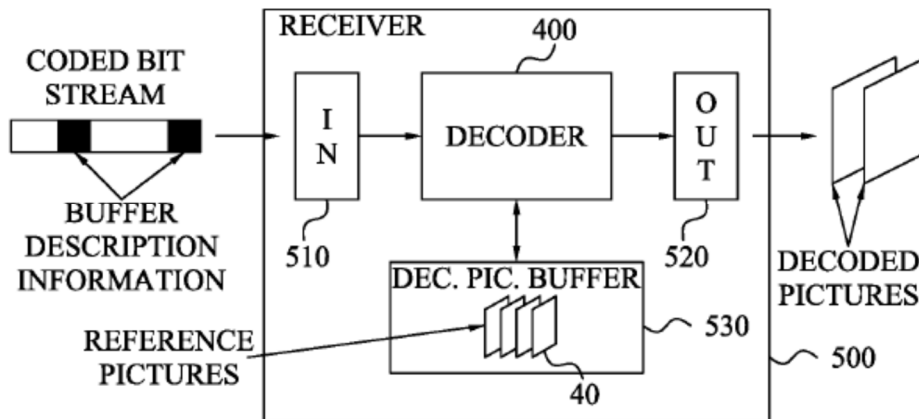
“The delta identifier(s) is(are) used in step S41 together with a picture identifier of the current picture to calculate the picture identifier(s) of the reference picture(s).”

Id. at 16:21-23.

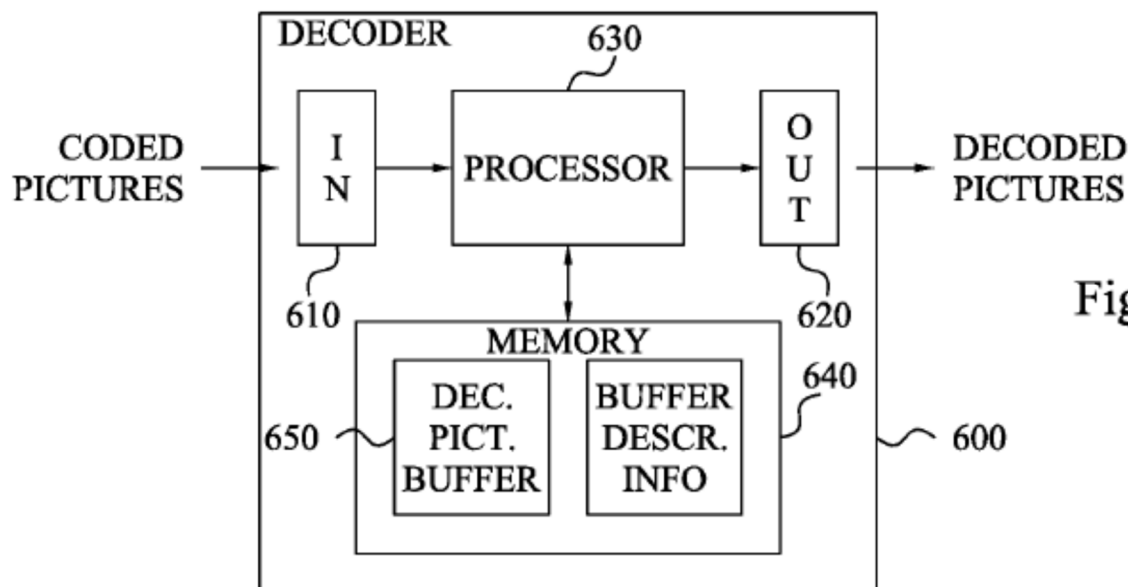
171. An example of the decoding process is depicted in Figures 9 and 10 of the '618 Patent:



172. A receiver with a decoder that implements the process is depicted in Figure 20:



173. Figure 22 of the '618 Patent depicts the decoder:



174. In addition to describing how the claimed invention identifies reference pictures for the current picture, the '618 Patent also describes how to identify reference pictures for a subsequent picture, i.e. a picture that will be decoded later in the bitstream. For example, it discloses that “[a]lternatively, or in addition, at least

one reference picture determined in step S1 could be used as encoding reference for a subsequent picture of the video stream, i.e. a picture to be encoded and decoded after the current picture.” *Id.* at 7:6-10. It further states that “[t]his subsequent picture is thereby after the current picture according to a decoding order (and encoding order).” *Id.* at 7:10-12.

175. The ’618 Patent further discloses that “[i]n an embodiment, the generation of nonexistent pictures is performed by the decoder prior to the picture decoding process of the picture containing the buffer description information.” *Id.* at 19:8-11. In this embodiment, “Step S72 compares the at least one picture identifier determined from the buffer description information with picture identifiers associated with reference pictures already stored in the decoded picture buffer.” *Id.* at 19:12-16. The ’618 Patent states that, “[i]f a determined picture identifier is not found among the picture identifiers of the reference pictures stored in the decoded picture buffer the picture associated with the determined picture identifier is determined to be missing or nonexistent.” *Id.* at 19:16-20.

176. In a further embodiment, “a non-existing picture is optionally generated in step S73 and stored in the decoded picture buffer. This non-existing picture is then assigned any parameters, such as picture identifier, obtained from the buffer description information with regard to this particular picture.” *Id.* at 19:20-25.

177. In the Annex, the '618 Patent also presents proposed syntax and semantics for carrying out the claimed invention. Exemplary syntax is reproduced below:

	Descriptor
slice_header() {	
first_tb_in_slice	ue(v)
entropy_slice_flag	u(1)
if(!entropy_slice_flag) {	
slice_type	ue(v)
pic_parameter_set_id	ue(v)
frame_num	ue(v)
additional_picture_id	ue(v)
if(!IdrPicFlag)	
idr_pic_id	ue(v)
if(pic_order_cnt_type == 0)	
pic_order_cnt_lsb /*	u(v)
if(!IdrPicFlag){	
buffer_description_reference_flag	u(1)
if(buffer_description_reference_flag == 1){	
buffer_description_id	ue(v)
}	
if(buffer_description_reference_flag == 0){	
number_of_explicitly_signaled_pictures	ue(v)
for(number_of_explicitly_signaled_pictures){	
additional_picture_id	ue(v)
if(num_reorder_frames > 0)	
delta_POC_sign	u(1)
absolute_delta_POC_minus_one	ue(v)
temporal_id	u(v)
}	
}	
}	
if(slice_type == P slice_type == B) {	
num_ref_idx_active_override_flag	u(1)
if(num_ref_idx_active_override_flag) {	
num_ref_idx_l0_active_minus1	ue(v)
if(slice_type == B)	
num_ref_idx_l1_active_minus1	ue(v)
}	
}	
ref_pic_list_modification()	
ref_pic_list_combination()	
if(num_ref_idx_l0 != 0)	

-continued	
	Descriptor
dee_ref_pie_marking()	
if(entropy_coding_mode_flag && slice_type != I)	
cabac_init_idc	ue(v)
slice_qp_delta	se(v)
if(adaptive_loop_filter_enabled_flag)	
alf_param()	
if(deblocking_filter_control_present_flag) {	
disable_deblocking_filter_idc	
if(disable_deblocking_filter_idc != 1) {	
slice_alpha_c0_offset_div2	
slice_beta_offset_div2	
}	
}	
if(slice_type == B)	
collocated_from_l0_flag	u(1)
} else	
if(entropy_coding_mode_flag && slice_type != I)	
cabac_init_idc	ue(v)
}	

Ex-1001 at 29:32-30:41.

178. Key to the claimed invention is the patent’s addition of the syntax element “absolute_delta_POC_minus_one.” A POSITA would have understood that POC stands for Picture Order Count which is an absolute identifier of a reference picture. Ex-1001 at 7:54-56. As explained in the patent, the delta POC value, i.e. a difference between the reference picture identifier and the current picture identifier, may be inserted into the bitstream rather than the POC itself to be more bit efficient. Ex-1001 at 9:32-54. A POSITA would have further understood that the purpose of the “minus one” is also to be more efficient as it is not necessary to signal a delta POC of zero.

C. The ’618 Patent’s Prosecution History

179. I have reviewed the prosecution history of the ’618 Patent.

180. The U.S. application that resulted in the '618 Patent was filed on June 13, 2019.

181. The Examiner filed a non-final rejection on December 12, 2019. Ex. 1002 at 96-106. This rejection noted that the Examiner was interpreting one or more of the filed claims as a means-plus-function claim under 35 U.S.C. § 112(f) or pre-AIA § 112, sixth paragraph. *Id.* at 98-99. The Examiner also rejected certain claims under nonstatutory double patenting. *Id.* at 99-100. Claims 17-19 were rejected under 35 U.S.C. § 101, and claims 8-16 were rejected under 35 U.S.C. § 112.

182. The Examiner also rejected the claims as obvious under 35 U.S.C. § 103 in view of Borgwardt and Lu. The Examiner asserted that “Borgwardt discloses all the claimed elements of claim 1 but Borgwardt does not appear to explicitly disclose retrieving, based on said buffer description information.” *Id.* at 104. However, the Examiner asserted that Lu does teach that element and it would have been obvious “to modify Borgwardt to incorporate the teachings of Lu to effectively manage memory space for storing pictures.” *Id.* at 105.

183. For claims 2 and 3, the Examiner similarly found that Borgwardt in view of Lu discloses the claimed methods. For claims 5, 6-10, 12, 14-15, and 17-18, the Examiner found that the claims are rejected for the same reasons as claims 1-3. For claims 4, 6, 11, 13, 16, and 19, the Examiner objected to them as dependent on

a rejected base claim, but found they “would be allowable if rewritten in independent form.” *Id.* at 105.

184. The Applicant responded to the non-final rejection on February 11, 2020. *Id.* at 123-38. In addition to certain claim amendments, the Applicant filed a terminal disclaimer to overcome the non-statutory double patenting rejection. The Applicant also argued that claims 8-16 should not be interpreted under § 112(f) or pre-AIA § 112, sixth paragraph; that claims 17-19 are patent eligible under § 101; that claims 8-16 are definite; and that the claims are patentable over the Examiner’s cited references.

185. With respect to the obviousness rejection, the Patentee distinguished Borgwardt as disclosing the use of modified MMCO commands for reference picture management, stating that “[n]one of these values define buffer description information that define a plurality of reference pictures.” *Id.* at 135. The Patentee also argued that the “reference picture selection layer [of Borgwardt] fails to disclose or suggest a delta identifier, a picture identifier, or a calculation of a reference picture identifier.” *Id.* at 136.

186. The Patentee also argued that Borgwardt’s disclosure of an absolute difference of picture numbers (ADPN), “which represents the absolute difference between the picture number of the currently re-mapped picture and the prediction value for that picture number,” does not “disclose or suggest a delta identifier

comprising a difference between the reference picture identifier and a picture identifier identifying the picture, a picture identifier identifying a picture, or calculation of a reference picture identifier, much less the above cited recitations of Claim 1.” *Id.*

187. The Patentee further argued that “[t]he sliding window and adaptive memory control buffering types [of Borgwardt] fails to disclose or suggest a delta identifier comprising a difference between the reference picture identifier and a picture identifier identifying the picture, a picture identifier identifying a picture, or calculation of a reference picture identifier, much less the above cited recitations of Claim 1.” *Id.* at 136-37. The Patentee made the same argument for why “[t]he control operation to be applied and the MMCO commands” of Borgwardt do not disclose the delta identifier limitations of the claims, much less the remaining claim limitations.

188. On April 6, 2020, the Examiner issued a Notice of Allowance without any further office actions. Ex. 1002 at 143-49. The Examiner noted that “[t]he substance of the applicants’ remarks, filed on 02/11/2020, are persuasive.” *Id.* at 148.

189. I have also reviewed the prosecution history of related patent application 15/299,997, which issued as 9,807,418 and was attached to the Petition as Ex-1009. I have summarized only the relevant parts of this file history below but reserve the right to supplement these opinions.

190. On January 26, 2017, the Examiner issued a non-final rejection alleging, among other things, that the claims were obvious over Borgwardt in view of Lu and Koyabu (US 20100238822 A1). Ex. 1009 at 302-12.

191. On April 24, 2017, Applicant responded to the rejection. Applicant argued that Borgwardt is “[i]n sharp contrast” to the claimed invention because it “describes various proposals for modifying MMCO storage management commands.” *Id.* at 346. Applicant argued these modified MMCO commands, which were sent in Borgwardt with other information, did not constitute “buffer description information defining at least one reference picture.” *Id.*

XI. DEVELOPMENT OF THE H.265 STANDARD

192. As the '618 Patent states, the invention was conceived in the context of the development of the H.265 Standard. Ex-1001 at 1:36-44. The original draft of H.265 employed the same reference management techniques as H.264. *Id.* at 20:28-39. However, the inventors of the '618 Patent submitted a proposal to the JCT-VC in July 2011 (the “F493 proposal”) suggesting a fundamental change to the way reference pictures were handled. Ex-2004. This fundamental change was to incorporate the invention of the '618 Patent into the H.265 Standard.

193. The inventors' F493 proposal—titled “Absolute signaling of reference pictures”—suggested describing reference pictures “in an absolute manner in the slice header of a picture instead of in a relative manner (through MMCO and sliding

window process).” Ex-2004 at 1. The inventors’ contribution described the invention as “improv[ing] error resilience, simplify[ing] temporal scalability and enabl[ing] decoding at a minimum number of decoded picture buffer slots.” *Id.*

194. In addition to recognizing the problems with relative signaling as to error resilience and temporal scalability, the inventors recognized how H.264 already used reference picture list reordering commands in every slice for B slices with gaps in frame number. *Id.* at 5. The inventors recognized that this use of reference picture list reordering commands was a problem, stating that “[t]he signaling of reference picture list modifications will increase the bit overhead of each slice header.” *Id.*

195. Given these problems with H.264 and the then-existing draft of the H.265 Standard, the inventors proposed adding “explicit signaling of the reference pictures, using a Buffer Description, in each slice header, containing information to deduce POC and temporal_id of all reference pictures marked as ‘used for reference’ in the DPB.” *Id.* at 6. They further proposed to “[r]emove all MMCO commands as the functionality provided by them is covered by Buffer Descriptions,” and “[r]emove the sliding window operation as this functionality is provided by Buffer Descriptions.” *Id.*

196. The inventors further specified that the proposal would “use POC for identification of pictures during reference picture list construction and decoded reference picture marking.” *Id.* at 6.

197. The contribution states:

It can be noted that the proposal implies conceptual changes to the decoding process. In H.264/AVC and current design of HEVC, operations are given to the decoder either implicitly (sliding window) or explicitly (MMCO) and the decoder is responsible of [sic] applying these operations and keeping track of the reference pictures (i.e. which pictures can be used for reference). In the proposed scheme the reference pictures (i.e. which pictures can be used for reference) is [sic] signaled within the current picture (in the slice header) thus removing the need of implicitly and explicitly signaled operations. This means that each picture will have an absolute description of the reference pictures instead of a relative description as in H.264/AVC (where delta-information is retrieved from MMCO or from the standard text, i.e. sliding window process).

Id.

198. As is common in the standardization process, a working group was formed to implement the changes proposed by the inventors. Ex-2005. While there were additions and changes during the working group process, the fundamental ideas of buffer description information and using an absolute reference picture identifier were retained in the final submission to the standardization group. Ex-2005.

199. These ideas from the F493 contribution were not only described and claimed in the '618 patent, but they were also ultimately adopted into the H.265

standard. Ex-2007. For example, the F493 proposal provided the key idea of using buffer description information, or a reference picture set, in every slice header to explicitly signal which reference pictures should be in the decoded picture buffer. As another example, the use of delta identifiers as part of that buffer description information was present in the original F493 contribution and made it into the standard. Both of these concepts are part of the claimed inventions of the '618 Patent.

200. Consistent with the F493 proposal, in the new reference picture management scheme of H.265, the decoder still maintains a DPB. H.265, however, does not use MMCO commands or sliding window.

201. Instead of MMCO commands or sliding window, the DPB is managed in H.265 based on what is called a “Reference Picture Set” or “RPS.” Ex-2007. The RPS is an absolute and explicit list that identifies every reference picture that is to be used for decoding the current picture and every picture that is to be used for subsequent pictures. The RPS is sent to the decoder in the header of every slice. When the decoder receives the RPS, it updates its DPB to match the contents of the RPS. Reference picture list(s) for the current picture are then constructed with *only* the reference pictures that are necessary to decode the current picture. Ex-2007 at § 8.3.2.

202. The RPS in the H.265 Standard is the same concept as “Buffer Descriptions” in the F493 contribution or “buffer description information” in the

'618 Patent. The H.265 Standard uses the exact language the patent uses to describe “buffer description information” for its definition of RPS. *Compare* Ex-1001 at 8:45-51 (“The buffer description information can therefore be regarded as information identifying a buffer description that is a set of reference pictures associated with a current picture. It consists of all reference pictures that are prior to the current picture in decoding order and that may be used for inter prediction of the current picture or any picture following the current picture in decoding order.”) *with* Ex-2007 at 29 (“[R]eference picture set (RPS): A set of reference pictures associated with a picture, consisting of all reference pictures that are prior to the associated picture in decoding order, that may be used for inter prediction of the associated picture or any picture following the associated picture in decoding order.”).

XII. ASSERTED REFERENCES

A. Mulroy (Ex-1004)

203. U.S. Patent Application Publication No. US 2009/0052531 (“Mulroy”) is titled “Video Coding” and was published on February 26, 2009.

204. As I will discuss in more detail below, Mulroy does not disclose or render obvious the '618 Patent. Instead, Mulroy still requires and relies upon use of preexisting reference picture management techniques from H.264, including the use of MMCO commands to maintain the DPB. Thus, Mulroy still suffers from the

problems identified by the '618 Patent. The '618 Patent is thus a substantial departure from Mulroy. The following describes Mulroy in more detail.

205. As set forth in the Abstract, Mulroy is directed to error recovery or error concealment techniques in H.264 coding. In particular, Mulroy explains that “[a] video encoder using inter-frame differential coding has a store (3) for containing a plurality of reference pictures.” Ex-1004 at Abstract. “Control means (10) select which locally decoded pictures are to be entered into, and removed from, as that the store always contains at least one relatively older picture that lies within a defined time window and [at] least one relatively younger picture that is more recent than the time window.” *Id.* According to Mulroy, “when [the encoder] receives an error message from a receiver, it switches the coder from its normal a state (in which it is allowed to choose any picture as predictor) into a state in which it is permitted to choose only a said relatively older picture for prediction.” *Id.*

206. Mulroy’s claimed method is consistent with the Abstract and involves “receiving error messages and in response thereto suppressing the use of relatively younger reference picture(s) as predictor and to choose only a said relatively older picture for prediction.” Ex-1004 at Claim 1. It further describes that, in its method, “said reference pictures are stored in a buffer having a storage capacity less than the round trip time to the decoder divided by the frame period.” *Id.*

207. Mulroy still requires use of preexisting reference picture management techniques from H.264. In Mulroy’s “normal” state, the decoded picture buffer is controlled by MMCO commands that are sent from the encoder to the decoder: “messages are sent using the ‘memory management control operations’ defined in H.264,” and “[t]he decoder buffer follows the encoder buffer by acting on these memory management control operations that the encoder has put in the bitstream.” *Id.* at [0025].

208. It also describes the situation where “the decoder loses a set of MMCO commands,” creating a “mismatch between the contents of the encoder and decoder buffers.” *Id.* at [0027]. In this scenario, Mulroy suggests making “use of the ‘reference picture list reordering’ commands defined by H.264” as an error handling message. *Id.* This discussion of reordering commands (which Mulroy also refers to as “remapping instructions”) spans just three paragraphs ([0027]-[0029]). The rest of Mulroy is dedicated to describing the error recovery techniques that are discussed in the abstract and claimed.

209. In the three paragraphs that discuss remapping instructions, Mulroy states that “[t]he encoder may optionally include in the bitstream a set of remapping instructions that alter this default order into some other order, known as the remapped order.” *Id.* Mulroy teaches using the same reordering commands from H.264 as its “remapping instructions,” explaining that “[r]emapping instructions are

defined as an explicit mapping from the frame number to remapped index (using frame number prediction for efficiency).” *Id.*

210. Mulroy explains how these remapping instructions may signal to the encoder that there is an error, stating, “[t]hese remapping instructions can be made use of in the present context for indicating to the decoder which frames (by their frame numbers) should be in its buffer.” *Id.* The way Mulroy does this “is for the encoder to insert remapping instructions in every slice (NAL unit) that reference all frames in its buffer (there is not necessarily any changing of the default order) and for the decoder to use these to determine which frame to remove from its buffer when it otherwise does not have the required knowledge—perhaps because there is no MMCO removal command because the encoder has fewer frames in its buffer, or because an MMCO removal command refers to a frame the decoder does not have.” *Id.* at [0028]. Notably, this is the entirety of Mulroy’s disclosure on this point.

211. Mulroy then explains, “we see that these remapping instructions are used by error recovery means operable to receive further messages from a coder, the further messages specifying which pictures should be in the store, and arranged in the event that a decoded picture is to be stored and a deletion message has not been received, to select a picture to be deleted from the store by reference to one of the further messages.” *Id.* at [0029].

212. In other words, Mulroy describes using remapping instructions to detect whether there is a mismatch between the encoder's buffer and the decoder's DPB. If there is a mismatch, the remapping instructions are then "used by error recovery means operable to receive further messages from a coder." *Id.* In that case, it is the "further messages" that specify "which pictures should be in the store" and "select a picture to be deleted from the store"—**not** the remapping instructions. *Id.* Mulroy never explains what these "further messages" are.

213. As an alternative to the further messages approach, Mulroy states that "the decoder could be fitted with error recovery means" that use "the same algorithm as used at the coder for selecting pictures to be deleted." *Id.* at [0029].

214. The remainder of Mulroy focuses on techniques for error detection and handling when remapping instructions are not used.

215. With respect to reference picture buffer management, Mulroy provides four "rules for reference buffer management:"

- 1) Do not remove a frame unless the buffer is full
- 2) Always keep the two most recent frames, i.e. ages 1 and 2
- 3) Subject to the $2m+2$ limit, always keep the oldest frame
- 4) Keep sufficient frames that when the oldest frame ages beyond the 'old' window there is still a frame in this 'old' window. This implies that a frame shall not be removed from the buffer if the time difference between the next oldest frame in the buffer and the next youngest frame in the buffer exceeds the size of the 'old' window (i.e. m). Where, as

here, the intermediate and ‘old’ windows are the same size, this requires at least one frame always in the intermediate window.

Id. at [0047]-[0051].

216. While these four rules are for the encoder’s buffer, a POSITA would know that, generally, rules applied at the encoder are also applied at the decoder where possible, because a video decoder generally attempts to mirror the corresponding video encoder.

B. H.264 (Ex-1005)

217. There are multiple editions of the H.264 standard. I understand the Petition relies on the edition that was purportedly published in November 2007, attached as Ex-1005. For the purposes of this declaration, my citations to the H.264 standard are to this edition, unless stated otherwise.

218. The way that reference picture signaling works in the H.264 standard is described in the ’618 Patent and in Paragraphs 75-139 of this report above.

C. Raveendran (Ex-1006)

219. U.S. Patent Application Publication No. U.S. 2006/0165176 (“Raveendran”) is titled “Method and apparatus for encoder assisted-frame rate up conversion (EA-FRUC) for video compression” and was published on July 27, 2006. Ex. 1006. Raveendran is directed to “an encoder assisted frame rate up conversion (EA-FRUC) system that utilizes video coding and pre-processing operations at the

video encoder to exploit the FRUC processing that will occur in the decoder in order to improve compression efficiency.” Ex. 1006 at Abstract.

220. I understand that the Petition relies on Raveendran for the physical components of a decoding system, such as a processor.

D. Dr. Creusere’s Incorrect Characterizations of H.264 and Mulroy, Including His Examples

221. In my opinion, Dr. Creusere mischaracterizes H.264 and Mulroy, and their proposed combination, on several occasions.

222. For example, Dr. Creusere asserts that reordering instructions can add pictures to the reference picture list(s) and remove pictures from the reference picture list. Ex. 1003 ¶ 63. This is not true.

223. H.264 reordering instructions cannot add a picture to a reference picture list or the DPB. Rather, pictures are added to the DPB and reference picture list as part of the normal decoding process. Ex-2003 at 106-07. Reordering can only then be used to change the order of the reference picture list for the current picture. Ex-2003 at 109 (“These commands are initiated by the syntax element `ref_pic_list_reordering_flag` and change the list order for the current slice only.”) (emphasis omitted).

224. H.264 reordering instructions also cannot remove a picture from a reference picture list or the DPB in H.264 and Mulroy. Rather, the only mechanism for removing a picture from the DPB is using MMCO commands or the sliding

window. Ex-2003 at 107. It is also important to note that, in H.264, all pictures in the DPB are in the reference picture list.

225. As another example of what I believe to be mischaracterisations, Dr. Creusere alleges that MMCO commands can indicate which pictures in the DPB will be used for decoding subsequent pictures. Ex. 1003 ¶ 70. This is also incorrect. As set forth in Table 7-9, MMCO commands can only indicate, for example, that a picture should be marked as a long-term picture. In H.264, there is no guarantee or indication that a long-term picture will or should be used to decode a subsequent picture. The mere fact of an encoder marking a picture as a long-term picture in an H.264 bitstream does not place any restrictions on which picture(s) may contain blocks that use this long-term picture for prediction. For example, within the context of H.264, it is perfectly valid for an encoder to mark a long-term picture, use it for predicting blocks in the current picture, then remove the picture from the DPB without ever using it to predict blocks in a subsequent decoded picture.

226. As another example of what I believe to be mischaracterisations, Dr. Creusere provides a set of examples purported to relate to Mulroy and H.264 in which the initial DPB is ordered as (ST149, ST138, ST144, LT12, LT17). Ex. 1003 ¶¶ 117-144. These examples are incorrect in the context of H.264.

227. First, these examples are not disclosed anywhere in Mulroy or H.264. Instead, they are manufactured by Dr. Creusere based on assumptions he makes after having the benefit of reviewing the '618 Patent.

228. Second, Dr. Creusere's examples merge the ideas of reference picture lists and the DPB into one concept. However, as explained above, the reference picture lists and DPB are separate and distinct. For example, reference picture lists must be generated for each slice or picture (i.e. they are not persistent lists that are used in the decoding of multiple pictures). The reference picture lists do not themselves actually include any pictures. The reference picture lists, instead, include an indexed list of pointers to the location of pictures in the DPB. In the case of H.264 long-term reference pictures, the reference picture lists indicate a further relative indicator such as "LT1" which in turn points to the location of a picture in the DPB, assuming that the necessary H.264 MMCO command(s) have been correctly received. In contrast, the DPB provides a persistent storage of pictures that can be used in the decoding and display of pictures. In H.264, the DPB is maintained using the sliding window and MMCO commands. Further, in H.264, there is no mechanism for reordering the DPB. The DPB and reference picture lists thus serve fundamentally different purposes.

229. Third, pictures in H.264 are always ordered according to the default rules, which Dr. Creusere appears to ignore. Under the rules of H.264, the default

order of pictures in a reference picture list of the pictures identified by Dr. Creusere is (ST149, ST144, ST138, LT12, and LT17).¹

230. Fourth, long-term picture indices of 12 and 17 are in my opinion and experience unrealistically large. The numbers associated with long-term pictures are indices, and do not indicate or depend on the order of the picture in display or decoding order. In other words, a long-term picture index number represents the index to which a long-term picture is assigned—starting at 0. There can only be as many long-term picture indices as there are slots in the DPB (the number of reference frames, Ex-1005 at 89), and even then, it would be highly unlikely in my experience for every picture in the DPB to be a long-term picture. In Dr. Creusere's examples, the DPB holds up to 5 pictures. And in real world applications of H.264, a typical size of DPB is around 4 pictures.²

231. Fifth, Dr. Creusere's third example of combining Mulroy and H.264 (Paragraphs 135-144) is in my view misleading for another reason. Dr. Creusere

¹ Dr Creusere provides these examples in the context of his 'Mulroy-H.264/AVC Combination'. Mulroy states that it uses H.264's 'Flexible Macroblock Ordering' technique (Mulroy [0030]) which can only be used in H.264's Baseline or Extended Profiles, neither of which supports B-pictures. Thus reference pictures in this proposed combination must therefore be organised according to H.264's P-picture reference picture ordering rules, which I summarise in Paragraphs 72, 119, and 138-9 above.

² Example: Consider HD video encoded using H.264, at a Level of 4.1. Ex 1005 (H.264) at 283. The frame size of HD video (1920x1080 pixels) expressed in macroblocks (16x16 pixels) is 8100 macroblocks, which is within the Level limit of 8192 macroblocks per frame. Each frame can be stored in uncompressed form using $1920 \times 1080 \times 1.5 = 3110400$ bytes, assuming 4:2:0 colour sub-sampling. The maximum DPB size (MaxDPB) is $12288 \times 1024 = 12582912$ bytes, i.e. slightly more than 4 frames of uncompressed HD video.

implies that Mulroy's method is necessary for the decoder to know that short-term reference picture 149 should be in its DPB after decoding (Paragraph 144). However, the decoder would have already known that short-term reference picture 149 should be in its DPB after decoding, without the need for Dr. Creusere's interpretation of Mulroy; that is the default H.264 case, i.e. if the picture with frame_num 149 is expected but not received, any H.264 decoder is capable of identifying that an error has occurred and take appropriate action.

232. Finally, Dr. Creusere's need to perform a lengthy analysis and manipulation of H.264 and Mulroy to arrive at his examples, based on sparse text in Mulroy (Paragraphs 27-29), is evidence that this is not how a POSITA would have understood Mulroy prior to the '618 Patent.

233. For example, Mulroy states only that "every slice ... contains a message informing the decoder as to exactly which frames should be in the decoder buffer at that time" and that "[o]ne way of doing this is to make use of the reference picture reordering commands defined by H.264", such that "these remapping instructions can be made use of... for indicating to the decoder which frames (by their frame numbers) should be in its buffer." Ex-1004 at 27.

234. From this limited disclosure, Dr. Creusere extrapolates that Mulroy actually discloses to a POSITA (1) certain specific parameters sent in an H.264 bitstream with particular values in a particular order (such as

reordering_of_pic_nums_idc and abs_diff_pic_num_minus1) (Ex-1003 at ¶¶ 119-124), (2) including indications of long_term_pic_num (*Id.* at ¶¶ 122-123), even though H.264's long_term_pic_num does **not** indicate any frame by its frame number, contrary to the cited disclosure in Mulroy (Ex-1004 at [0027]). *See, e.g.*, discussion *supra* at ¶¶ 90, 98-103 and 108-12, 126.

235. In a similar way, from a single sentence in Mulroy Paragraph 28, Dr. Creusere extrapolates that Mulroy actually discloses to a POSITA a series of 5 steps required for an encoder to signal a series of specific syntax values in a certain order, followed by a decoder selecting to remove a specific long-term reference frame. Ex-1003 at ¶¶ 129-133 (citing Ex-1004 (Mulroy) at [0028] (“The way we do this is for the encoder to insert remapping instructions in every slice (NAL unit) that reference all frames in its buffer (there is not necessarily any changing of the default order), and for the decoder to use these to determine which frame to remove from its buffer when it otherwise does not have the required knowledge”)).

236. The specific steps of these and other ‘examples’ provided by Dr. Creusere are nowhere disclosed in H.264 or Mulroy and in my view there is simply no way a POSITA could have reached the outcomes and conclusions proposed by Dr. Creusere without using the ’618 Patent as a roadmap.

237. I address further issues with Dr. Creusere’s examples in the element-by-element discussion below.

XIII. THE CHALLENGED CLAIMS OF THE '618 PATENT ARE NOT OBVIOUS

A. Petitioner's Ground 1 Fails Because Claims 1-7 Are Not Obvious In View Of Mulroy And H.264

238. The following addresses examples of why certain limitations of claims 1-7 are not obvious in view of Mulroy and H.264.

239. While I address certain limitations to demonstrate that each of the challenged claims are not unpatentable, I reserve the right to supplement this declaration to include additional information and opinions related to any and all challenged claims and claim elements.

1. The prior art does not disclose or render obvious “buffer description information defining a plurality of reference pictures,” as required by claims 1-7.

240. Claims 1 and 5 require “retrieving, from the encoded representation of the picture, buffer description information defining a plurality of reference pictures.” Claims 2 through 4 depend from claim 1, and claims 6 and 7 depend from claim 5, and thus also include this limitation.

241. Dr. Creusere argues that “Mulroy-H.264/AVC renders this element obvious because Mulroy discloses it.” Ex-1103 at ¶ 147. I disagree. Neither Mulroy, H.264, nor any combination of the two disclose or render obvious “buffer description information defining a plurality of reference pictures.”

a. Dr. Creusere does not apply the plain and ordinary meaning of “buffer description information defining a plurality of reference pictures”

242. Dr. Creusere's opinions seem to imply that he has interpreted the requirement of "buffer description information defining a plurality of reference pictures" as merely something describes the state, or even part of the state, of a buffer.

243. This interpretation ignores the plain language of the claims. The claims require buffer description information *defining a plurality of reference pictures*. Thus, under the plain language, it is not enough to just identify information that describes the buffer. The buffer description information must define reference pictures. A type of signaling that describes the state of a buffer, such as the DPB, but does so relative to something else, such as a prior state of the DPB and/or a previously received MMCO command, does not define a reference picture. Thus, Dr. Creusere's interpretation of the claims is contrary to the plain language of the claims.

244. Dr. Creusere's interpretation is contrary to the intrinsic record. For example, Dr. Creusere's interpretation is so broad that buffer description information could theoretically encompass MMCO commands. In fact, Dr. Creusere has suggested that the MMCO commands in H.264 are buffer description information. Ex-1003 ¶ 103. The invention of the '618 Patent, however, proposes to remove the use of MMCO commands in favor of an absolute and explicit buffer description information or RPS.

245. Thus, to a POSITA, the phrase “buffer description information defining a plurality of reference pictures” requires an RPS, or an absolute and explicit form of reference picture signaling (i.e. independent from previously decoded pictures). This is how a POSITA would have understood this claim language in view of the specification and prosecution history.

246. **First**, the claim requires that the buffer description information “*define* a plurality of reference pictures.” This indicates to a POSITA that the buffer description information must describe with particularity the reference pictures. And the fact that the term is “*buffer* description information” indicates to a POSITA that the information must define the reference pictures in the buffer, not in some other aspect of the codec (such as a reference picture list).

247. **Second**, the specification confirms my understanding of the plain and ordinary meaning of the term “buffer description information.” In explaining the problem that the ’618 Patent was intended to resolve, the ’618 Patent explains that “[i]f the slice headers of the picture contain Memory Management Control Operations (MMCO) commands, adaptive memory control process is applied after decoding of the picture to obtain *relative* reference to the pictures to be stored in the decoded picture buffer; otherwise a sliding window process is applied to obtain *relative* reference to the pictures to be stored in the decoded picture buffer.” Ex-1001 at 2:5-11 (emphasis added).

248. The '618 Patent expressly sought to solve the problems associated with this relative reference, and thus proposed the use of buffer description information that its used for reference pictures in an “absolute and explicit way”:

In clear contrast to the prior art solutions in which correct reference picture management is dependent on that previously encoded pictures have been correctly received and decoded, *the embodiments provide buffer description information that is used for reference pictures in an absolute and explicit way instead of a relative or implicit way.* Thus, the encoded representation of a picture contains the information about which reference pictures to use for reference during decoding independent of the encoded representations of previous pictures in the video stream.

Ex-1001 at 4:39-48 (emphasis added).

249. Thus, the buffer description information “is independent of the encoded representations of previous pictures in the video stream.” Ex-1001 at 4:39-48. The '618 Patent further explains that “the buffer description identified or obtained by the buffer description information enables an unambiguously[sic] identification of the relevant reference picture(s).” *Id.* at 7:62-64. The specification repeats this understanding of “buffer description information” as being “absolute” and “explicit” at least nine more times. *Id.* at 6:38-48; 6:57-64; 7:38-46; 12:27-35; 14:20-24; 15:5-13; 20:40-44; 20:45-49; 25:43-48.

250. The specification also makes this clear by explaining that “buffer description information” and “Reference Picture Set (RPS)” are used synonymously.

Id. at 8:22-31. A POSITA would have understood that a “Reference Picture Set (RPS)” represents the absolute and explicit boundaries of which reference pictures are in the decoded picture buffer.

251. **Third**, this plain meaning is further confirmed by the provisional application that resulted in the '618 Patent, which I understand is a part of the patent's disclosure. The provisional application explains that “[a] problem with the prior art solutions is that correct decoder reference picture management is dependant [sic] on that previously decoded pictures have been correctly received and decoded,” but that the proposed invention solves this problem “by specifying which decoded pictures to be used for reference pictures in an absolute (explicit) way instead of a relative (implicit) way.” Ex-2008 at 18. The provisional describes making the following changes to previous signaling methods, among others: (1) “[a]dd explicit signaling of the reference pictures (Buffer Description) in each slice header, containing POC and temporal_id of all reference pictures available for prediction in the DPB”; (2) “[r]emove all MMCO commands”; and (3) “[r]emove the sliding window operation.” *Id.* at 18. The provisional further explains “[i]n the proposed scheme the reference pictures (i.e. which pictures can be used for reference) is signaled within the current picture (in the slice header) thus removing the need of implicitly and explicitly signalled relative operations.” *Id.* at 19 (referring to MMCO commands as “explicit” but still “relative”). According to the provisional

application, “[t]his means that each picture will have an absolute description of the reference pictures instead of a relative description as in H.264/AVC (where delta-information is retrieved from MMCO or from the standard test, i.e. sliding window process.” *Id.*

252. ***Fourth***, the plain and ordinary meaning is confirmed by the prosecution history. In particular, the prosecution history makes clear that “buffer description information defining a plurality of reference pictures” ***excludes*** relative or implicit types of reference picture signaling, such as the use of MMCO commands. As explained during prosecution, MMCO commands are simply numerical values and “[n]one of these values define buffer description information that define a plurality of reference pictures.” Ex-1002 at 135.

253. This is true regardless of whether a reference modifies the signaling to use MMCO commands and some other signaling method. For example, the Applicant distinguished Borgwardt during prosecution as not disclosing buffer description information because it used modified MMCO commands for reference picture signaling. The Examiner accepted this distinction when issuing the patent, even though Borgwardt discloses sending other types of information along with the MMCO commands—including for example, a “Re-Mapping of Picture Numbers Indicator (RMPNI),” “Absolute Difference of Picture Numbers (ADPN),”

“Difference of Picture Numbers (DPN),” “Long-term Picture Index (LPIN),” and “Maximum Long-Term Picture Index Plus 1 (MLIP1).” Ex-2012 at 3, 8.

254. Thus, a POSITA reading this claim limitation in view of the patent’s disclosure and prosecution history would have understood that “buffer description information defining a plurality of reference pictures” requires an RPS, or an absolute and explicit form of reference picture signaling (i.e. independent from previously decoded pictures) instead of a relative or implicit form of reference picture signaling (i.e. dependent on previously decoded pictures).

b. Mulroy in view of H.264 still requires the use of relative or implicit signaling instead of absolute and explicit signaling

255. As explained below, the reference picture signaling disclosed in Mulroy and H.264 does not use “buffer description information defining a plurality of reference pictures.” Instead, the signaling is all relative or implicit, and therefore cannot be the claimed buffer description information.

256. It is unclear precisely which elements of Mulroy that Dr. Creusere and the Petition argue are the “buffer description information defining a plurality of reference pictures.” For example, Dr. Creusere points to sets of MMCO commands as the buffer description information in one section of his declaration (Ex-1003 ¶ 103) but the reordering commands in a different section (Ex-1003 ¶ 148). Dr. Creusere also refers to a “message” that uses “reordering commands” as the buffer

description information, which I understand to mean some broader interpretation of message that includes both MMCO commands and reordering commands. Ex-1003 ¶ 152. Dr. Creusere elsewhere points to the reference picture indices as “defining a plurality of reference pictures,” even though he does not explicitly say the indices are the buffer description information. These inconsistent citations to different elements of Mulroy/H.264 as disclosing the claimed buffer description make understanding the Petition and Dr. Creusere’s theory on this limitation unclear at best.

257. Regardless of which argument the Petition and Dr. Creusere are making, this limitation is not met by any of the elements in Mulroy/H.264 for the reasons discussed below.

**(a) MMCO Commands are not the claimed
buffer description information**

258. Dr. Creusere implies that H.264 MMCO commands are “buffer description information defining a plurality of reference pictures.” Ex-1003 ¶ 103. I disagree. MMCO commands are not the claimed buffer description information at least because they are a form of relative signaling—not absolute. MMCO commands can only carry out limited operations relating to a picture in the decoded picture buffer, for example removing a picture from the DPB (by marking it as “unused for reference”) or converting a short-term reference picture to a long-term reference picture. Ex-1005 at 86-89 (§ 7.4.3.3, Table 7-9). These commands are relative, or

dependent on the previous state of the DPB, because they indicate to the decoder what actions to take with respect to what is already in the DPB (i.e. previously decoded pictures).

259. MMCO commands are the exact type of relative signaling the '618 Patent replaced with its absolute signaling (i.e. buffer description information defining a plurality of reference pictures). *See, e.g.*, Ex-1001 at 9:8-10 (“This approach of the embodiments provides a significant advantage over the relative reference picture signaling of the prior art using MMCO command.”); *id.* at 20:28-44 (“This means that each picture will have an absolute description of the reference pictures instead of a relative description as in H.264/MPEG-4 AVC where delta information is retrieved from MMCO or from using the sliding window process”).

260. Dr. Creusere himself agrees that MMCO commands are a form of relative signaling. Ex-1003 at ¶ 54 (“MMCO values in the Adaptive Memory Control approach of the H.263 standard use what the '618 patent describes as ‘relative’ references—references that are interpreted relative to correctly decoded previous pictures.”). Lenovo’s expert during the ITC investigation, Dr. Hu, likewise agreed that MMCO commands are a form of relative signaling. Ex. 2011 at 882:15-18.

261. Because H.264 MMCO commands are relative, they cannot be the claimed buffer description information.

262. However, even if the claimed buffer description information did not need to be absolute (which it does for the reasons discussed above), the claim language still requires the buffer description information to “*defin[e]* a plurality of reference pictures.” MMCO commands do not define reference pictures. MMCO commands are an instruction from the encoder to the decoder regarding how to change the existing state of the DPB. Thus, MMCO commands cannot be the claimed buffer description information for this additional reason.

**(b)Reordering commands are not the claimed
buffer description information**

263. Dr. Creusere appears to argue that the reordering commands of H.264, as used in Mulroy, are the claimed buffer description information. Ex-1003 ¶¶ 148-49, 154-55. I disagree. Reordering commands—even as used in Mulroy or as interpreted by Dr. Creusere—are not the claimed buffer description information. As explained above in Paragraphs 117-28, reordering commands (reference picture reordering instructions) include information about the order of the reference picture list, not the decoded picture buffer. Moreover, the reordering commands are just another form of relative signaling.

264. Mulroy describes using the reordering commands “defined in H.264” as “remapping instructions.” Reordering commands “are initiated by the syntax element `ref_pic_list_reordering_flag` and change the list order for the current slice only.” Ex. 2003 at 109 (original emphasis omitted). These commands are relative

because they inform the decoder what order to put the reference pictures in for the current reference picture list *relative* to their position in the initial reference picture list.

265. Reordering commands only work if the initial reference picture list is correct. The initial reference picture list is the default order of the available reference pictures in the DPB (not an arbitrary order as Dr. Creusere seems to suggest, *see* discussion *supra* at ¶¶ 138-39). Thus, reordering commands depend on the DPB being correct, which is itself dependent on previously decoded pictures. This makes reordering commands inherently relative.

266. Dr. Creusere and the Petition imply that using reordering commands in “every slice” somehow makes the relative reordering commands absolute. *See, e.g.*, Ex-1003 at ¶¶ 112, 148, 150-51, 157-58. This is not so. Mulroy discloses using the same, unmodified reordering commands specified in H.264, and those reordering commands are relative for the reasons explained above. Using a relative process more frequently cannot change the fact that the process is still relative.

267. The relative nature of the reordering commands, even as used by Dr. Creusere, is especially apparent when considering long-term reference pictures. Dr. Creusere provided no examples to explain what happens in the Mulroy/H.264 combination when an MMCO command assigning a short-term to a long-term

picture is lost. This is telling. As the '618 Patent points out, an H.264 decoder cannot adequately handle lost MMCO commands that affect long-term pictures. Ex-1001 at

268. In H.264 and Mulroy, long-term pictures are given a long-term picture index value. This index value is assigned to a picture by a MMCO command. As I explained above in Paragraphs 109-12, 152, and 162, a long-term picture index value is a relative indicator whose correct interpretation (i.e. which picture data the index value corresponds to) depends on previously received MMCO commands. If the MMCO command associated with a long-term picture is lost, there is no way for the decoder to recover without clearing the DPB (for example by receiving an IDR picture). In Dr. Creusere's examples, the decoder would receive reordering commands that indicate there is a long-term picture that does not exist in the decoder's DPB. But without the lost MMCO command, the decoder does not know which data in its DPB is supposed to correspond to that long-term picture. Thus, the reordering commands can at most indicate that there may be an error—but they cannot fix the error, i.e. they cannot unambiguously identify the reference pictures required to decode the current picture. This example further demonstrates that the H.264 reordering commands as used in Mulroy are relative. If they were absolute, then they would in theory be able to replace MMCO commands and avoid this type of error altogether. This is not what Mulroy teaches, and Dr. Creusere's examples do not solve this shortcoming with H.264 reordering commands.

269. To illustrate the relative nature of the signaling in Dr. Creusere's examples, we need only go back one picture with respect to his Example 1 (Creusere 117-124) and consider the effect of a lost MMCO command. Assume we are decoding the previous frame, ST148, which is a P picture.

270. Before decoding ST148, the DPB would look like this: ST144, ST138, ST137, LT12.

271. Now, assume that an MMCO command is sent as part of coded short-term picture 148, indicating to the decoder to assign the picture that was decoded with picture number 137, currently stored in the DPB as the short-term picture ST137, to long-term picture index LT17.

272. Thus, after decoding ST148, the DPB **should** be: ST148, ST144, ST138, LT12, LT17.

273. Now assume, however, that the MMCO command sent with ST148 was lost or corrupted. In this case, the DPB before decoding frame 149 would be: ST148, ST144, ST138, ST137, LT12.

274. Using Mulroy's method as interpreted and extended by Dr. Creusere, the decoder may be able to detect an error because it receives reordering commands that tell it the DPB should be: ST148, ST144, ST138, LT12, LT17.

275. However, the decoder has no knowledge that LT17 should correspond to the picture that was decoded with picture number 137. In fact, it has no

information that indicates **any** data in its DPB that corresponds to LT17, originally picture number 137. Thus the decoder may detect that there is an error but **not** what actual picture data is or should be in the DPB. In other words, the decoder has not received information “about which reference pictures to use for reference during decoding independent of the encoded representations of previous pictures in the video stream”. In fact, Dr. Creusere’s decoder has to discard ST137 (which is no longer identified as being in the DPB and according to Dr. Creusere will be marked as unused for reference) and cannot decode any image blocks that use LT17 for prediction, since it has no knowledge of any picture data corresponding to LT17. This problem will persist until the next IDR, just as the ’618 Patent explains. Ex-1001 at 7:62-64. This example alone demonstrates that Dr. Creusere’s proposed combination does not provide the claimed Buffer Description Information.

276. Consider the following further example. Assume that the decoder is about to start decoding the P picture with frame number 104. The DPB at the start of decoding comprises ST103, ST101, ST100 and ST99 (short-term pictures identified by frame number) and LT0, LT3 (long-term pictures identified by the relative index LongTermPicNum). The default order of the reference picture list (list0) is as illustrated below.

Current picture: frame_num 104 

Decoded Picture Buffer before decoding 104:

Index:	Identifier:	Picture:
0	ST 103	
1	ST 101	
2	ST 100	
3	ST 99	
4	LT 0	
5	LT 3	

277. Assume that the current frame (104) contains an MMCO command of type 3, which assigns the picture with frame number 100 (formerly ST100) as a long-term reference picture with identifier LT1. Thus the relative index LT1 should point to the picture data decoded as frame 100. The result of this MMCO command is as follows:

Decode frame_num 104
MMCO command 3: move ST100 to LT1
DPB after decoding and applying MMCO:

Index:	Identifier:	Picture:
0	ST 104	 (new)
1	ST 103	
2	ST 101	
3	LT 0	
4	LT 1	 (previously ST100)
5	LT 3	

MMCO3



278. What happens if the MMCO command is lost or corrupted? Because the MMCO command creates a *relative* identifier (LT1), subsequent decoded frames will have incorrect DPB information and cannot be successfully decoded, whether or not Dr. Creusere’s interpretation of Mulroy is applied.

279. First, consider an H.264 decoder without Mulroy’s method. Because the MMCO command is lost, ST100 remains identified as a short-term reference picture and LT1 is not assigned. When the decoder constructs the list0 reference picture list, the original ST100 and the long-term picture LT0 are in the wrong order as illustrated below. Subsequent decoded frames will have incorrect DPB information (for example, no LT1, LT0 persisting in the wrong position in the ordered reference index list). Thus, subsequent frames cannot be decoded correctly

because the decoder’s reference picture management is dependent on correct receipt of previous encoded pictures (in this case, the MMCO command associated with picture 104):

Index:	Identifier:	Picture:
0	ST 104	 (new)
1	ST 103	
2	ST 101	
3	ST 100	 (wrong position, ST not LT)
4	LT 0	 (wrong position)
5	LT 3	

280. Second, consider Dr. Creusere’s interpretation of Mulroy. According to Dr. Creusere, the decoder can use received remapping instructions (H.264’s reference picture list reordering instructions) to determine that the DPB should contain ST104, ST103, ST101, LT0, LT1 and LT3. However, at least two problems remain because of the relative signalling in H.264, namely (1) the picture with frame number 100 is not present in the reference index list and (2) the decoder has no information about which decoded picture should correspond to LT1 (illustrated below).

Index:	Identifier:	Picture:
0	ST 104	 (new)
1	ST 103	
2	ST 101	
3	LT 0	
4	LT 1	 (picture does not exist)
5	LT 3	

281. Subsequent decoded frames will have incorrect DPB information (for example, LT1 persisting in the reference index list, with no associated decoded picture data). Thus subsequent frames cannot be decoded correctly because the decoder’s reference picture management is dependent on correct receipt of previous encoded pictures (in particular, the MMCO command associate with picture 104). For example, consider a later point in decoding of the same sequence. Assuming that the maximum DPB size is 6 and that no further changes have been made to the long term reference pictures, the DPB now looks like the following. Reference index 4 continues to point to “LT1” which has no corresponding decoded picture data. Once again, the current picture cannot be correctly decoded:

Decoder DPB using Mulroy's method:

Index:	Identifier:	Picture:
0	ST 106	 (latest)
1	ST 105	
2	ST 104	
3	LT 0	
4	LT 1	 (picture does not exist)
5	LT 3	

282. Thus this example demonstrates once more that Dr Creusere's and Petitioner's best effort to construct a method based on Mulroy and H.264 fails to resolve a fundamental problem identified by the inventors of the '618 patent, namely the loss of an MMCO command that affects long term reference pictures ('618 at 2:56-3:2). This is for the same reason that the inventors identified, namely that reference picture management in H.264, in Mulroy and in Dr Creusere's combination, is dependent on previously encoded pictures having been correctly received and decoded ('618 at 4:39-42). In other words, the reordering commands are a relative form of signaling and therefore cannot be the claimed buffer description information.

283. However, as with MMCO commands, even if the claimed buffer description information did not need to be absolute, it still must “*defin[e]* a plurality of reference pictures.” The H.264 reordering commands, even as used in Mulroy and

by Dr. Creusere, do not define reference pictures. Reordering commands represent the change in position from the initial reference picture list to the current reference picture list. If they “define” anything, it is just that—a change (or lack of change) in position—they do not define the reference pictures themselves. Further, when applied to a long term reference frame index such as LT1, H.264’s reference picture reordering commands merely denote a change in position of a relative index (LT1) which itself relies on previously received MMCO commands to point to a reference picture. Thus, Mulroy’s use of the reordering commands cannot constitute the claimed buffer description information for this additional reason.

(c) The “message” is not buffer description information

284. Dr. Creusere argues that an ambiguous and undisclosed “message” in Mulroy that uses reordering commands is the claimed buffer description information. Ex-1001 ¶¶ 150-52. I disagree. A message that uses reordering commands and other types of relative signaling, such as MMCO commands, cannot be the claimed buffer description information. The combination of two or more relative processes does not make an absolute one. A set of information that includes relative signaling, such as MMCO commands, cannot be absolute. At least part of the process will always depend on previously decoded pictures. Thus, the combination of reordering commands and MMCO commands into a broader “message” means that the “message” is not absolute and cannot be the claimed buffer

description information. If it were absolute, it would not need to make use of relative signaling. In any event, such a combination of commands is nowhere disclosed in the cited references.

285. Even if a form of relative signaling could satisfy the claimed buffer description information, the broader “message” allegedly sent in Mulroy (but not described by Mulroy) does not “defin[e] a plurality of reference pictures.” Its components, presumably including MMCO commands and/or reordering commands, describe other things—not the reference pictures themselves.

**(d)Reference picture indices are not buffer
description information**

286. Dr. Creusere argues that reference picture indices define a plurality of reference pictures, thereby implying that these indices are the claimed buffer description information. ¶ 153. I disagree. Reference picture indices are not the claimed buffer description information and do not define a plurality of reference pictures.

287. Reference indices are not an absolute and explicit form of signaling. Reference indices are a selection of one or two “slots” in the DPB. These “slots” or “indices” may be associated with any reference picture. Reference indices are sent with a coded block such as a macroblock, selecting one or two specific reference pictures out of the set of available reference pictures already in the DPB. The syntax element refidx is sent per inter-coded block. Thus, not only are these indices

dependent on the previously decoded pictures having been correctly received and decoded, but they also are not an absolute identification of all the pictures used for reference, merely an index into a list or an array (the reference picture list).

288. Even if reference picture indices were absolute, they do not “defin[e] a plurality of reference pictures” as Dr. Creusere and the Petition suggest. Which reference picture is associated with which reference index changes as more pictures are decoded and therefore added and removed from the DPB (and reference picture list). Consider the below example from my book. Index 0 in list 0 corresponds to frame 150 at first, but after frame 151 is encoded, index 0 then corresponds to frame 151 and frame 150 moves to index 1.

Reference Picture Order Example: P slices (list0)

The reference picture list is initially empty. The current *frame_num* is 150 and the maximum size of the DPB is 5 frames. Italics indicate a *LongTermPicNum*. Notice that by default, the first entry in list0 is the most recently coded frame. This is likely to be the ‘best’ prediction reference for the majority of macroblocks.

Operation	list0(0)	list0(1)	list0(2)	list0(3)	list0(4)
Initial state	-	-	-	-	-
Encode frame 150	150	-	-	-	-
Encode 151	151	150	-	-	-
Encode 152	152	151	150	-	-
Encode 153	153	152	151	150	-
Encode 154	154	153	152	151	150
Encode 155	155	154	153	152	151
Assign 154 to LongTermPicNum 3	155	153	152	151	3
Encode 156 and mark it as LongTermPicNum 1	155	153	152	1	3
Encode 157	157	155	153	1	3

(e) H.263 is irrelevant

289. Finally, only in the background section of his declaration, Dr. Creusere argues that “absolute” references were used in H.263 (and ultimately incorporated

into H.264) because they use a “Temporal Reference for Prediction (TRP)” syntax element. Ex-1002 at ¶ 54. I disagree. Neither Dr. Creusere nor the Petition explicitly argue that the TRP syntax element is the claimed buffer description information. Nor could they, at least because H.263’s TRP syntax element can only be used to identify a single reference picture, not a plurality of reference pictures.

290. For example, Annex N of H.263 states that the encoder can identify *one* reference picture by TRP (“TRP indicates *the* Temporal Reference which is used for prediction of the encoding”, Ex 1011 at 99, emphasis added. Thus, TRP cannot be buffer description information, at most it is a single reference picture identifier.

291. In addition, there is no updating the DPB (“additional picture memory”) based on buffer description information or a reference picture identifier in H.263. *See* N.5 (showing that the decoder buffer only operates in sliding window mode, “correctly decoded video picture segments shall be stored into memory for use as later reference pictures on a first-in, first-out basis”, Ex 1011 at 102).

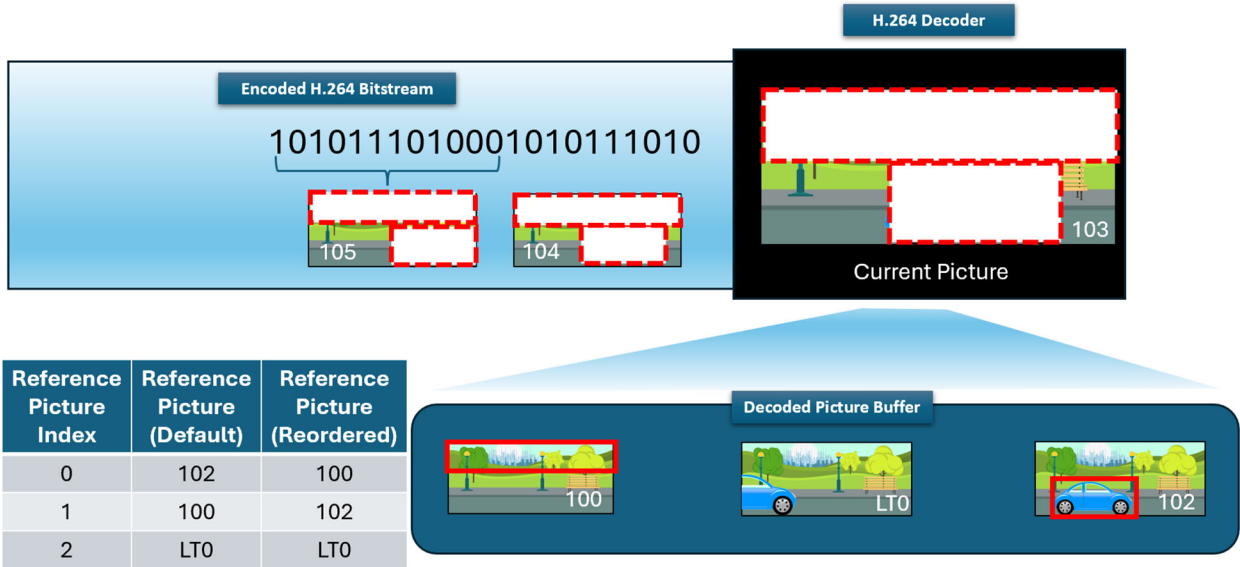
2. ***The prior art does not disclose or render obvious “a reference picture identifier that identifies a respective reference picture among the plurality of reference pictures, the respective reference picture comprising a decoding reference for the picture” as required by claims 1-7.***

292. A POSITA would have understood that this claim language requires identifying a picture as a decoding reference for the current picture.

293. Dr. Creusere summarily argues that this limitation is met because Mulroy discloses using reordering commands to arrive at a frame number of reference picture and that the frame number “identifies a reference picture among the reference pictures usable to decode the current picture.” Ex-1003 at ¶¶ 156-58.

294. However, H.264’s reordering commands do not identify *which* pictures are used as a decoding reference for the current picture. Instead, as explained at length above, the reordering commands describe the change in order of *all* reference pictures that *should* be in the DPB. In H.264, not all pictures in the DPB are necessarily used for decoding the current picture, and a decoder cannot determine based on the reordering commands which pictures in the DPB are used for decoding the current picture and which are not.

295. Consider our example from earlier:



296. The reference picture reordering instructions may indicate to the decoder that its DPB should have Pictures 100, 102, and LT0, but nothing in the reordering instructions tells the decoder which of these pictures are used as decoding references for the current picture—if any of them are used as decoding references at all.

297. As another example, it is possible that, for a particular picture being decoded, the DPB includes exclusively long-term pictures, and none of those pictures are being used to decode the current picture. Thus, reordering instructions identifying those long-term pictures would not identify any reference picture for the current picture. Accordingly, the reordering instructions and reference pictures list do not provide any indication as to what pictures are used for the current picture and which are used for a subsequent picture.

298. To illustrate a simple example that is consistent with the H.264 standard, consider an encoder that produces a bitstream conforming to H.264, in which Picture 100 is identified as a P-picture (using syntax element `slice_type` in each slice header, H.264 at 7.4.3). Because Picture 100 is a P picture, the decoder constructs a reference picture list, `list0`, referencing all pictures present in the DPB as explained earlier in Paragraphs 72, 119, and 138-39. Assume that the encoder chooses to encode each macroblock of Picture 100 as an Intra macroblock. This is entirely consistent with the H.264 standard, since the encoder can choose Inter or

Intra prediction for each macroblock in a P-picture³. In this example, (1) there is a reference picture list for Picture 100, indexing each picture in the DPB, but (2) no part of Picture 100 is predicted using any pictures in the DPB. Thus, while the reordering instructions might identify changes to the reference picture list, the instructions do not provide any indication as to which pictures should be used as a reference picture for the current frame or another frame.

299. Thus, this limitation is not met or rendered obvious by Mulroy or the combination of Mulroy with H.264.

3. *The prior art does not disclose or render obvious “updating a decoded picture buffer based on the reference picture identifier,” as required by claims 1-7.*

300. In just one paragraph, Dr. Creusere argues that Mulroy discloses updating the DPB based on the picture number of a reference picture that is determined using remapping instructions. Ex-1001 at ¶ 164. I disagree.

301. First, Mulroy references the “frame number,” not the “picture number.” Ex-1004 at [0027]. A POSITA would have understood that “frame number” and “picture number” are two different values in the H.264 standard, which is the standard Mulroy applies. The “frame number” syntax element in H.264 is denoted “frame_num” and appears in the main syntax table of Section 7.3.3 (Slice header

³ An encoder might make such a decision for example in order to handle a problematic scene change in a video sequence, where a suitable inter prediction source cannot be found in previously encoded frames.

syntax). Ex-1005 at 46. The H.264 standard defines “frame_num” as “an identifier for pictures” that is represented by certain values in the bitstream. *Id.* at 80-81. The “picture number” value in H.264 is “PicNum” or “pic_num” and does not appear in the main table in 7.3.3, but is referenced in 7.3.3.1 (Reference picture list reordering syntax). *Id.* at 47 and assigned as a variable in § 8.2.4.1. Decoding process for picture numbers (*Id.* at 113). That frame number and picture number are distinct variables with distinct meanings in the H.264 standard is evident from Section 8.2.5 (decoded reference picture marking process), for example, which provides that “[a] short-term reference picture is identified for use in the decoding process by its variables FrameNum and FrameNumWrap and its picture number PicNum.” *Id.* at 119.

302. Second, even if the “frame number” of Mulroy were the same as the “picture number,” Mulroy does not teach using the frame number to update the DPB. Rather, Mulroy describes using reordering commands to detect an error, and once an error is detected, then using “error recovery means operable to receive further messages, the further messages specifying which pictures which pictures should be in the store.” Mulroy at [0029].

303. “Alternatively,” Mulroy describes “using the same algorithm as used at the coder for selecting pictures to be deleted.” *Id.* Mulroy explains the “algorithm as used at the coder” later in the patent publication by defining its four “rules of buffer management.” *Id.* at [0047]-[0051].

304. Thus, Mulroy describes either updating the DPB with “further messages” or updating the DPB with the “algorithm as used at the coder.” Neither are the frame number.

305. Not only does Mulroy not disclose using the frame number to update the DPB, but the frame number in Mulroy *cannot* be used to update the DPB. In Mulroy, the reordering commands proposed as a form of “backup” to enable a decoder to detect an issue with lost MMCO commands. At the point an error is detected, the DPB is already out of sync. Ex-1004 at [0027] (“If the decoder loses a set of MMCO commands, there will be a mismatch between the contents of the encoder and decoder buffers.”). The encoder learns this from the reordering commands because it sends commands directing the decoder to order reference pictures in a certain way, but the decoder can recognise that it does not have those reference pictures in its DPB. Ex-1004 at [0027] (“These remapping instructions can be made use of in the present context for indicating to the decoder which frames (by their frame numbers) *should* be in its buffer.”). At this point, Mulroy must use other error recovery means (aka “further messages” or the encoder’s algorithm) to update the DPB. It cannot use the frame numbers to update the DPB because the reference pictures having those frame numbers have already been inadvertently removed from the DPB because of the lost MMCO commands. Ex-1004 (explaining the decoder “otherwise does not have the required knowledge—perhaps because there is no

MMCO removal command because the encoder has fewer frames in its buffer, or because an MMCO removal command refers to a frame the decoder does not have.”). As explained previously, once a reference picture is removed from the DPB, it cannot then be placed back into the DPB.

306. This is fundamentally different to the '618 Patent because the '618 Patent describes using buffer description information defining a plurality of reference pictures *instead* of MMCO commands and sliding window operation, thereby preventing the problem of lost or corrupted MMCO messages, which in turn prevents the incorrect decoding of a whole series of pictures. Ex-1001 at 4:39-54. Because the reference pictures with the correct reference picture identifier have not been inadvertently removed from the DPB, the '618 Patent is able to use the reference picture identifier to update the DPB in every instance.

4. ***The prior art does not disclose or render obvious “i) a picture identifier identifying a respective reference picture among the plurality of reference pictures as the decoding reference for the picture and ii) a picture identifier identifying a respective reference picture among the plurality of reference pictures as a decoding reference for a subsequent, according to a decoding order, picture in the video,” as required by claim 2.***

307. Notably, Dr. Creusere does not address the plain and ordinary meaning of Claim 2. Regardless, in my opinion, the plain meaning of Claim 2 requires two separate types of identification: (1) identifying a reference picture as the decoding reference for the current picture, and (2) identifying a reference picture as a decoding

reference for a subsequent picture. This how a POSITA would have understood the plain meaning of the claim language, especially given that there are two subparts and the claim language explicitly requires *identifying* a respective reference picture *as the decoding reference*.

308. One benefit of distinguishing which pictures are needed for the current picture versus which pictures are only in the DPB for subsequent pictures is that the decoder need not construct a reference picture list that includes every picture in the DPB. Instead, each reference picture list need only refer to the reference pictures that are necessary for decoding the current picture. This is more efficient, as it saves bits per inter coded block, as I explain earlier in Paragraph 157.

309. Mulroy does not disclose these requirements. The only thing the Petition and Dr. Creusere point to as allegedly meeting these requirements is the fact that H.264, the standard employed by Mulroy, discloses both short-term and long-term pictures. Ex-1001 at ¶¶ 168-70.

310. This reasoning is incorrect. The use of short-term and long-term pictures does not meet these limitations. In H.264, both short-term and long-term pictures are available for decoding the current picture *and* subsequent pictures. In the prior art, there is no way of identifying which short-term and long-term pictures are used or may be used for the current picture versus which short-term and long-term pictures are used or may be used for a subsequent picture. For example, in

H.264, it is possible that a long-term picture in the DPB is used for a current picture but then never used for a subsequent long-term picture. Moreover, even if a long-term picture will be used as a reference picture for a subsequent picture, there is no way to determine the connection between a subsequent picture and that long-term picture from the reordering commands. In other words, using H.264 there is no way to identify a long-term picture as a reference picture for a particular subsequent picture.

311. In H.264, all DPB pictures (short-term and long-term) must be included in the current picture's reference picture list, for every decoded picture. This means that the decoder always includes all reference pictures from the DPB in its reference picture lists, even though some of those pictures may not be necessary for decoding the current picture. This may be inefficient compared with the invention of the '618 as I explain earlier in Paragraph 157.

312. Moreover, as described above in Paragraphs 294-98, there is no mechanism for identifying which reference pictures are to be used for the current frame or a particular subsequent frame using H.264's reference picture reordering commands. This illustrates a fundamental difference between buffer description information that explicitly identifies a reference picture and the relative signaling of Mulroy and/or H.264.

313. Notably, Lenovo's position on this limitation has evolved since the ITC case. Lenovo's expert during the ITC proceeding, Dr. Hu, presented the theory that the identifying a picture as a reference picture for a subsequent picture limitation was met because the decoder could deduce which pictures in the DPB would be used for a subsequent picture. This theory was incorrect for all the reasons I discussed at the ITC. For example, Dr Hu proposed during her testimony at the hearing, that an encoder could send fewer reordering instructions than there were pictures in the DPB. I understand that her proposal was that any pictures not identified in this way, should be used only for decoding subsequent pictures. Ex. 2011 at 836:19-837:9. However, according to Dr Hu's interpretation of Mulroy, the decoder would automatically **remove** any non-identified pictures from the DPB, making it impossible to achieve the result of identifying some pictures as used for the current picture and some pictures as used for decoding subsequent pictures.

314. Now, perhaps recognizing the failure of that position before the ITC, Lenovo's new expert, Dr. Creusere, relies solely on the theory that a long-term picture meets the "identifying a picture as a reference picture for a subsequent picture" limitation. But this theory, too, is incorrect for all the reasons discussed above. Petitioner Lenovo's shifting approach to this claim reinforces my view that Petitioner's argument as to why this limitation is met has no basis in the cited prior art.

315. For at least these reasons, the prior art does not disclose or render obvious Claim 2.

B. Petitioner's Ground 2 Fails Because Claims 8-19 Are Not Obvious In View Of Mulroy, H.264, and Raveendran

316. The Petition and Dr. Creusere add Raveendran to the combination in Ground 2 to argue that the physical components of a decoder (e.g. a processor) are disclosed. They do not rely on Raveendran to disclose any of the other limitations, such as the ones provided below, nor could they. Accordingly, the arguments discussed above for claim 1-7 apply equally to the claims below.

1. The prior art does not disclose or render obvious "buffer description information defining a plurality of reference pictures," as required by claims 8-19.

317. The limitation found in independent claims 1 and 5 (and therefore dependent claims 2-4 and 6-7), "retrieving, from the encoded representation of the picture, buffer description information defining a plurality of reference pictures," is also found in independent claims 8, 12, 14, and 17 (and therefore dependent claims 9-11, 13, 15-16, and 18-19). Thus, for the reasons stated above in Paragraphs 240-91, this limitation is not rendered obvious for claims 8-19.

2. ***The prior art does not disclose or render obvious “a reference picture identifier that identifies a respective reference picture among the plurality of reference pictures, the respective reference picture comprising a decoding reference for the picture” as required by claims 8-19.***

318. The limitation found in independent claims 1 and 5 (and therefore dependent claims 2-4 and 6-7), “a reference picture identifier that identifies a respective reference picture among the plurality of reference pictures, the respective reference picture comprising a decoding reference for the picture,” is also found in independent claims 8, 12, 14, and 17 (and therefore dependent claims 9-11, 13, 15-16, and 18-19). Thus, for the reasons stated above in Paragraphs 292-99, this limitation is not rendered obvious for claims 8-19.

3. ***The prior art does not disclose or render obvious “update a decoded picture buffer based on the reference picture identifier,” as required by claims 8-19.***

319. The limitation found in independent claims 1 and 5 (and therefore dependent claims 2-4 and 6-7), “update a decoded picture buffer based on the reference picture identifier,” is also found in independent claims 8, 12, 14, and 17 (and therefore dependent claims 9-11, 13, 15-16, and 18-19). Thus, for the reasons stated above in Paragraphs 300-06, this limitation is not rendered obvious for claims 8-19.

4. *The prior art does not disclose or render obvious “i) a picture identifier identifying a respective reference picture among the plurality of reference pictures as the decoding reference for the picture and ii) a picture identifier identifying a respective reference picture among the plurality of reference pictures as a decoding reference for a subsequent, according to a decoding order, picture in the video,” as required by claim 9.*

320. Claims 9 requires the same limitations required by Claim 2 and discussed in detail above with respect to Ground 1: “i) a picture identifier identifying a respective reference picture among the plurality of reference pictures as the decoding reference for the picture and ii) a picture identifier identifying a respective reference picture among the plurality of reference pictures as a decoding reference for a subsequent, according to a decoding order, picture in the video.” Thus, for the reasons stated above in Paragraphs 307-15, this limitation is not rendered obvious for Claim 9.

XIV. OBJECTIVE INDICIA OF NON-OBVIOUSNESS

321. In my view, there are many strong objective indicia of non-obviousness that support non-obviousness of the challenged claims.

322. Prior to the '618 Patent, the experts in the industry concluded that the best mechanism for managing reference pictures was to use MMCO commands and a sliding window. This functionality was adopted into the H.264 Standard and was proposed to be inserted into the H.265 Standard. As described in section XI, it was the inventors proposed a fundamental shift to remove MMCO commands in favor

of explicit reference picture signaling. Once the industry saw the elegant solution, it was quickly adopted into the H.265 Standard. This adoption is evidence of industry praise for the solution.

323. After the implementation of the inventions claimed in the '618 Patent, others in the industry copied the technology. H.265/HEVC adopted the claimed technology, and I understand that Lenovo's products utilize the claimed technology through their support for H.265/HEVC. The invention of the '618 Patent, garnered industry attention and praise. I understand the invention of the '618 Patent to have been adopted as part of the H.265/HEVC standard. In 2017, HEVC was honored with an Emmy at the 69th Engineering Emmy Awards. *See, e.g.*, Ex. 2013; Ex. 2014.

324. The '618 Patent also had unexpected results related to efficiency. A POSITA could well have assumed that the explicit approach of the claimed invention would have cost significantly more bits than using a default reference picture order and relying on successful delivery of MMCO commands and/or Supplemental Enhancement Information messages in the first instance, as is disclosed in the prior art (Ex-1001 at cols. 1-3). But this was not the case. Ex 2004 at 13-14.

325. It was also unexpected that using the claimed buffer description information to update the decoded picture buffer would solve problems such as lost MMCO messages. Prior art at the time taught away from this idea. For example, Mulroy taught that the decoded picture buffer should be updated based on aspects

such as timing of the pictures—not based on a reference picture set such as the claimed concept – in order to handle such problems.

XV. SUMMARY AND OTHER REMARKS

326. It is my opinion that Petitioner has failed to show that any of the Challenged Claims set forth in Grounds 1 and 2 are unpatentable as obvious as explained above.

327. My opinions expressed in this Declaration are based on the information available to me at this time. To the extent any additional information becomes available, I reserve the right to supplement my opinions contained herein.

I hereby declare under penalty of perjury under the laws of the United States
of America that the foregoing is true and correct.

Date: 7th March 2025



Iain Richardson, Ph.D.