UNITED STATES PATENT AND TRADEMARK OFFICE

_____

BEFORE THE PATENT TRIAL AND APPEAL BOARD

_____

ASUSTEK COMPUTER INC.,

Petitioner

v.

NOKIA CORP. AND NOKIA TECHNOLOGIES OY,

Patent Owner

_____

Case: IPR2025-01345

U.S. Patent No. 9,036,701

_____

DECLARATION OF DR. IAIN RICHARDSON

TABLE OF CONTENTS

I, Dr. Iain Richardson, declare as follows:

I. INTRODUCTION

1. I have been retained on behalf of Nokia Technologies OY and asked to provide opinions in respect of Inter Partes Review (IPR) proceeding IPR2025-01345 which relates to U.S. Patent No. 9,036,701, which I will refer to as the '701 Patent.

2. This Declaration summarises the independent opinions that I have formed to date which are based on the material currently available to me. I reserve the right to modify my opinions, if necessary, based on further review and analysis of information that I receive subsequent to the filing of this Declaration, including in response to any positions taken by Petitioners or their experts with which I have not yet been provided.

3. My findings, as explained below, are based on my study, experience, knowledge and background in the fields discussed below. I am informed by my education and my extensive experience in the fields of video compression and multimedia coding and communications. I have also relied on my review and analysis of the prior art, information provided to me and information I have independently reviewed.

4. I have previously provided three declarations concerning the '701 Patent subject to a protective order in the ITC investigation *Certain Video-Capable*

Laptop, Desktop Computers, Handheld Computers, Tablets, Televisions, Projectors, and Components and Modules Thereof, Inv. No. 337-TA-1448. I have re-used and adapted material from these previous declarations where I feel it is appropriate.

II. EDUCATIONAL BACKGROUND AND QUALIFICATIONS

5. Details of my professional qualifications and background are summarised below and are set out in detail in my Curriculum Vitae, attached to this Declaration as Exhibit 2010.

6. I am Director and co-founder of Vcodex BV (previously Vcodex Ltd).

7. I have over 30 years of experience related to video and image compression and coding. I am qualified by education and experience to testify as an expert with respect to subject matter in the fields of video, image, multimedia and general-purpose data encoding, decoding, compression, decompression, and transmission.

8. I received a Master of Engineering (M.Eng.) degree in Electronic and Electrical Engineering from Heriot-Watt University in Edinburgh, Scotland in 1990. I received a Doctor of Philosophy (Ph.D.) degree in Video Compression from Robert Gordon University in Aberdeen, Scotland in 1999. I worked as a Digital Signal Processing ("DSP") Hardware Designer with GEC Avionics Ltd. from 1990 to 1993.

9. 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 Robert Gordon University.

10. At Robert Gordon University, I founded and ran an image communication technology research laboratory. I carried out original research in the field of data, image, and video compression, initiated and managed research projects, and supervised research students.

11. I am the author of four books and over 70 journal and conference papers on video and image coding and communication, including two widely cited books on the H.264 / MPEG-4 industry standards for video compression. In my 2002 book, *Video Codec Design*, I discussed in detail the efficient implementation of image and video compression on software and hardware platforms.

12. 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. The designs included 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.

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. Onecodec's technology and business activities were merged with Vcodex Limited in 2015.

15. I have previously prepared expert reports and testified in a number of cases concerning video and image compression, graphics compression, general-purpose data compression, and multimedia streaming.

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

17. Because of my background, training, and experience, I am qualified as an expert in the technical areas relevant to the '701 Patent. My company Vcodex BV is being compensated for my independent analysis as an expert with respect to this IPR proceeding, but neither my compensation nor Vcodex BV's compensation is contingent in any way on the content of my analysis or the outcome of this

proceeding. I have no financial interest in Petitioners (Snap Inc. or Hisense USA Corporation) or Patent Owner (Nokia Technologies Oy) and I have no financial interest in the challenged patent.

III. MATERIALS REVIEWED

18. The opinions contained in this Declaration are based on the documents that I reviewed or cited here, including the Petition (as well as all materials and expert declaration cited there), my professional judgment, as well as my education, experience and knowledge regarding technologies related to the ’701 Patent at and around the time the patent was filed.

IV. LEGAL STANDARDS

1. Overview

19. I am not an attorney or a legal expert, and I do not opine on matters of law. The legal standards, which are listed in this section and which I applied in reaching the conclusions expressed throughout this Declaration, were provided to me and explained to me by counsel. In preparing this Declaration, I have endeavored to faithfully apply these legal standards. I do not intend to offer opinions on the law. Below, I describe the legal standards that I applied in analyzing the claims of the ’701 Patent.

20. I have been informed by counsel that the challenger of a patent’s validity, in this case the Petitioners, bears the burden of proving invalidity of the

patent.

2. Patent Claims

21. I have been informed by counsel that a patent's claims describe the invention made by the inventors and describe what the patent owner owns and what the owner may prevent others from doing.

22. I have been informed by counsel that claims may describe products, such as machines or chemical compounds, or processes for making or using a product. Claims are usually divided into parts or steps called limitations.

23. I have been informed by counsel that there are two types of claims: independent claims and dependent claims. An independent claim sets forth all the requirements that must be met in order to be covered by that claim. Thus, it is not necessary to look at any other claim to determine what an independent claim covers. A dependent claim does not itself recite all of the requirements of the claim, but refers to another claim for some of its requirements. In this way, a dependent claim depends on another claim. The law considers a dependent claim to incorporate all of the requirements of the claim to which it refers.

24. I have been informed by counsel that a dependent claim then adds its own additional requirements. To determine what a dependent claim covers, it is necessary to look at both the dependent claim and any other claims to which it refers.

25. I have been informed by counsel that in a case where a dependent

claim incorporates the limitations of an independent claim upon which it depends, if a reference fails to disclose any of the limitations of the independent claim, the dependent claims which depend on that independent claim cannot be made invalid by such a reference. As a result, any of the references at issue in this proceeding that fail to disclose all of the limitations of an independent claim necessarily also fail to disclose all of the limitations of claims which depend on that independent claim.

3. Burden of Proof for Invalidity

26. I have been informed by counsel that the party challenging the validity of an issued U.S. patent in an IPR bears the burden of proving invalidity through facts supported by a preponderance of the evidence. I have been informed that a "preponderance of the evidence" means evidence sufficient to prove that a fact is more likely true than it is not. I understand that, for the determination of whether to institute an IPR, the challenger must show a reasonable likelihood of success in invalidating at least one challenged claim.

4. Claim Construction

27. I have been informed by counsel and I understand that the first step in determining validity is to properly construe the claims. I have further been informed by counsel that, in this proceeding, the Board must determine the scope of the claims by using the same claim construction standard that United States District Courts use. Under this standard, the words of a claim are given their "ordinary" and "customary"

meaning as would have been understood by a person of ordinary skill in the art at the time of the invention, unless there has been some specific definition or disavowal of subject matter provided in the patent or file history. I have been informed that such a person of ordinary skill in the art looks to the claims, specification, and file history in understanding the meaning of the claim terms. I have applied this understanding in this declaration.

5. Person of Ordinary Skill in the Art

28. I have been informed by counsel and I understand that a person having ordinary skill in the art may be defined by factors such as, when considered at the time of the alleged invention: (a) the levels of education and experience of the inventor and other persons actively involved in the field; (b) the types of problems encountered in the field; (c) prior art solutions to those problems; (d) rapidity with which innovations are made; and (e) the sophistication of the technology.

6. Anticipation

29. I have been informed by counsel and I understand that a patent claim can be found invalid as anticipated if, and only if, each and every element of the claim, as properly construed, is found either explicitly or inherently in a single prior art reference. If the single prior art reference does not disclose explicitly or inherently any one claim limitation, then that claim is not anticipated. I have further been informed that the single prior art reference cannot merely disclose each

element; rather, it must disclose all of the elements as arranged in the claim. I have been informed that for a prior art reference to "inherently" disclose something, the reference must necessarily function in accordance with, or include the claimed limitations, to anticipate. The missing descriptive matter cannot merely probably or possibly include the claimed limitations. I have further been informed by counsel that, with respect to an anticipation analysis, it is improper to combine separate, distinct embodiments of a single prior art reference.

30. I have been informed by counsel that for a reference to be anticipatory, the reference must contain an enabling disclosure. I have been informed that merely naming or describing the subject matter is insufficient if it cannot be produced without undue experimentation.

7. Obviousness

31. I have been informed by counsel that a patent claim can be found invalid as obvious if the differences between the claim and the prior art references would have been obvious to a person of ordinary skill in the art at the time of the invention, taking into account (1) the scope and content of the prior art, (2) the differences between the prior art and the claims, (3) the level of ordinary skill in the art, and (4) any objective indicia of non-obviousness. I have further been informed by counsel that a claim can be found to be obvious in light of either a single prior art reference or the combination of multiple prior art references. I have been informed

that references qualify as prior art for an obviousness determination only when they are analogous to the claimed invention. Two separate tests define the scope of analogous prior art: (1) whether the art is from the same field of endeavor, regardless of the problem addressed; and (2) if the reference is not within the field of the inventor's endeavor, whether the reference still is reasonably pertinent to the particular problem with which the inventor is involved.

32. I have been informed that to be obvious in light of a single prior art reference or the combination of multiple prior art references, there must be a reason either to modify the single prior art reference or to combine the two or more prior art references to achieve the claimed invention. I have been informed by counsel that this reason may come from a teaching, suggestion, or motivation to combine which may come from the reference or references themselves, the knowledge or "common sense" of one skilled in the art, or from the nature of the problem to be solved, and may be explicit or implicit from the prior art as a whole. I have been informed by counsel that it is not sufficient that a person of ordinary skill merely could have modified or combined references, but that there must be a showing that a person of ordinary skill would have done so.

33. I have been informed by counsel that the combination of familiar elements according to known methods may be obvious when it does no more than yield predictable results.

34. I have further been informed by counsel that there is no suggestion to combine, however, if a reference teaches away from its modification or combination with another source. A reference may be said to teach away when a person of ordinary skill, upon reading the reference, would be discouraged from following the path set out in the reference, or would be led in a direction divergent from the path that was taken by the applicant, or if it suggests that the line of development flowing from the reference's disclosure is unlikely to be productive of the result or property sought by the applicant. If, when combined, the references would produce a seemingly inoperative device, then they teach away from their combination. I have been informed that it is improper to combine references where the references teach away from their combination. I have been informed by counsel that the degree of teaching away will depend on the particular facts.

35. I have been informed that if a proposed modification would render the prior art device unsatisfactory or inoperable for its intended purpose, then the proposed modification would generally not be obvious.

36. I have been informed by counsel that it is improper to use hindsight in making the obviousness determination.

37. I have been informed by counsel that some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness must be made. I have been informed that, for example, there may be no rational underpinning

to support the legal conclusion of obviousness where systems are already complete and the party challenging validity seeks to mix and match components, which each exist in the separate systems, from these complete systems.

V. PERSON OF ORDINARY SKILL IN THE ART (POSITA)

38. Based on my review of the Petition, I understand that Petitioners have identified that the POSITA "would have had a (1) Bachelor's degree in electrical engineering, computer engineering, computer science, or a comparable field of study such as physics, and (2) approximately two to three years of practical experience with video encoding/decoding . . . [and] [m]ore education could substitute for less experience, and vice-versa." For the purposes of my analysis for this Declaration, I have used the definition that the Petitioners have identified.

39. All of my opinions in this Declaration are from the perspective of a person of ordinary skill in the art, as the Petitioners defined it above and during the relevant time frame. The '701 Patent claims priority to U.S. Provisional Application No. 61/368,316, which was filed on July 28, 2010. By this date, I possessed at least the qualifications of such a POSITA.

VI. CLAIM CONSTRUCTION

40. I understand that claim terms are typically given their ordinary and customary meaning, as would have been understood by a person of ordinary skill in the art at the time of the alleged invention. In considering the meaning of the claims,

I also understand that one must consider the claim language, specification, prosecution history, and the knowledge of a person of ordinary skill in the art at the relevant time.

41. I understand that the Petition stated that no constructions are necessary and therefore has not proposed any terms for construction. Petition at 10.

42. Therefore, at this time, I have applied the plain and ordinary meaning as a POSITA would have understood it at the time of the invention and in accordance with the principles that I state above.

VII. BACKGROUND

1. The '701 Patent

43. U.S. Patent No. 9,036,701 (the "'701 Patent"), titled "Method and Apparatus for Providing Complexity Balanced Entropy Coding" issued to inventors Jani Lainema, Kemal Ugur, and Antti Hallapuro on May 19, 2015. The '701 Patent claims priority to U.S. Provisional Application No. 61/368,316, which was filed on July 28, 2010.

44. The '701 Patent is directed to a method and apparatus for "complexity balanced entropy coding … in order to provide a high level of capability with respect to video coding and decoding while keeping the costs in terms of complexity relatively low." '701 Patent at 1:54-60. Prior to the '701 Patent, the H.264 video coding standard employed two distinct coding engines, 1) "a low

complexity Context Adaptive Variable Length Coding based approach (CAVLC)" and 2) "the Context Adaptive Binary Arithmetic Coding approach (CABAC)." '701 Patent at 4:23-42. While CABAC provided improved efficiency in entropy coding, the '701 Patent notes that "the computational requirements of the CABAC method increase substantially as a function of the utilized bitrate" and "the CABAC approach may cause problems in real life implementations as the computational requirements of that operation increases." *Id*. In other words, H.264's CABAC offered superior compression efficiency over CAVLC but was too computationally complex to be utilized on certain consumer devices. This split between computational complexity led to CAVLC being the default entropy coding engine for low-complexity devices at the cost of increasing bitstream

sizes.

45. One goal of the H.265 standardization effort was to "achieve a balance between coding efficiency and practicality" through adjusting "the design of transform coefficient coding in HEVC." One way in which this was achieved was to "[i]ncrease the percentage of bypass bins and group bypass bins together for higher throughput." Transform Coefficient Coding in HEVC, Sole et al., IEEE Transactions on Circuits and Systems for Video Technology, Vol 22, No. 12, December 2012 at 1766. The inventions of the '701 Patent provide means to achieve this balance.

46. As the '701 Patent describes, "by categorizing syntax elements

differently, different categories of symbols may be operated on differently in order to achieve relatively higher efficiency in a lower complexity implementation." '701 Patent at 4:43-56. The '701 Patent provides an example of this process utilizing a CABAC coding engine, wherein "if encoding is performed using CABAC of Pipe/V2V related approaches, the corresponding decoding approaches for each may be implemented for the first category symbols. Meanwhile, syntax elements belonging to the second category may employ the same coding engine (e.g., the entropy coding engine 80) as first category symbols except that static fifty percent bin probabilities may be employed in relation to decoding of these symbols (e.g., by employing a bypass coding mode of a CABAC engine)." '701 Patent at 10:25-36.

47. The '701 Patent explains this is advantageous because "first category encoding may be structured to provide relatively high efficiency while second category encoding may be structured to provide lower complexity (e.g., by bypassing context updating)." '701 Patent at 10:37-54. The '701 Patent also acknowledges that not all syntax elements within a bitstream must be processed in this manner. This allows for flexibility in application as "the method and apparatus of on[e] example embodiment may encode some of the syntax elements in accordance with complexity balanced entropy coding and others of the syntax elements in accordance with other coding schemes, such as in accordance with arithmetic coding schemes." '701 Patent at 10:55-65. This means that "the

complexity balanced entropy coding of an example embodiment of the present invention may be utilized jointly or in combination with arithmetic or other coding schemes." *Id*.

48. The '701 Patent notes that "an instance of the categorizer 82 and the entropy coding engine 80 may be provided at the video encoder/decoder 48." '701 Patent at 10:66-11:4. In other words, the categorization may be performed at both the encoder and decoder to align with the entropy coding mechanisms employed. This categorization can also take place at the "block, slice, picture, or sequence levels." '701 Patent at 11:41-61.

49. The '701 Patent provides an exemplary embodiment in which the categorization is performed for DCT coefficient syntax elements: "As indicated in FIG. 4, by employing an example embodiment that categorizes syntax elements and then encodes each category differently to reduce complexity, the computational complexity as indicated by the number of DCT coefficients, is kept below the desired maximum level by utilizing context adaptive arithmetic coding while bypassing probability estimation for DCT coefficient data for some symbols." '701 Patent at 11:23-40. In other words, as more DCT coefficients are coded, the frequency of occurrence of syntax elements representing the coefficients also rises. This increase is a problem in video coding because e.g. if the motion and detail in video content increases, the number of non-zero coefficients increases. An increase in the number

of DCT coefficients "may cause problems in real life implementations", '701 Patent at 4:37-42. The '701 Patent provides a solution to balance the increase in complexity with compression efficiency through measuring the frequency of occurrence of syntax elements (such as syntax elements representing DCT coefficients) and adjusting entropy coding modes in response, making it possible to manage the computational complexity required to encode and decode data such as DCT coefficients. '701 Patent at 11:23-40, Fig. 4.

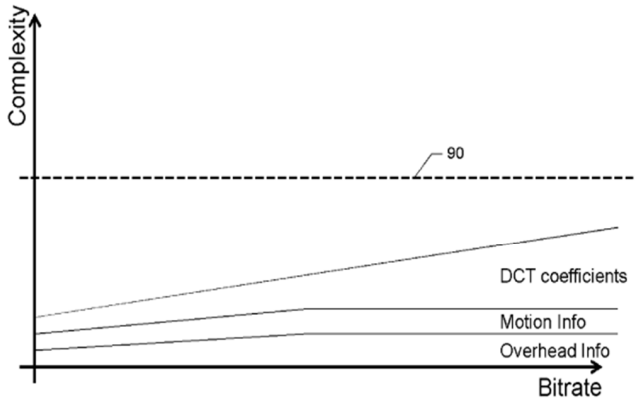

FIG. 4.

50. The '701 Patent provides a flowchart "of a method and program product according to an example embodiment of the invention" and notes that "each block of the flowchart, and combinations of blocks in the flowchart, may be implemented by various means, such as hardware, firmware, processor, circuitry and/or other device associated with execution of software including one or more computer program instructions." '701 Patent at 12:4-34.

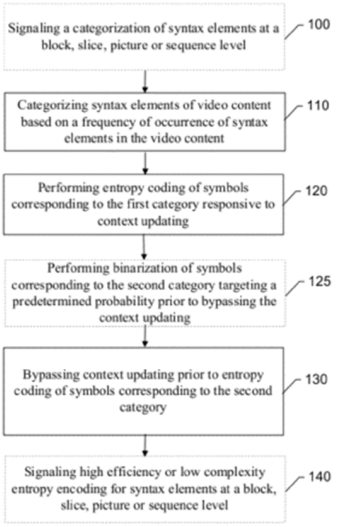

FIG. 5.

51. The ’701 Patent goes on to explain that one embodiment “may include categorizing syntax elements of video content based on a frequency of occurrence of syntax elements in the video content at operation 110. The categorizing may include defining at least a first category and a second category of symbols” and “may further include performing entropy coding of symbols corresponding to the first category responsive to context updating at operation 120, and bypassing context updating prior to entropy coding of symbols corresponding to the second category at operation 130.” ’701 Patent at 12:35-53. The ’701 Patent explains that “additional optional operations may also be included (examples of which are shown in dashed lines in FIG. 5).” ’701 Patent at 12:54-13:15.

52. In my opinion, Petitioner is incorrect to state that "[a]pplicant did not contest that Marpe '806 disclosed the categorization or the entropy limitations [of the '701 Patent]." Petition at 7. On the contrary, Applicant explained that Marpe '806 fails to disclose "categorizing a plurality of SEs. into first and second categories based on a frequency of occurrence of syntax elements in the video content." Ex-1002 ('701 File History) at 232-233, Apr. 30 2014 Response.

53. Applicant explained that Marpe is directed to sending distinct bins to either context updating or bypass coding "based only on the binary makeup of the bin." Ex-1002 ('701 File History) at 114, Dec. 2, 2014 Response. In other words, Applicant distinguished the '701 Patent over Marpe '806 by pointing out that Marpe '806 operates based on characteristics of the symbol (bin) values rather than the frequency of occurrence of syntax elements. As with every ground raised in this Petition, Marpe '806 does not categorize or process syntax elements based on the frequency of occurrence of syntax elements.

2. The Prior Art

a. Murashita (Ex-1007)

54. U.S. Patent No. 5,650,783 ("Murashita"), titled "Data Coding/Decoding Device and Method" was filed on Sep. 8, 1995. I agree with Petitioner that Murashita relates to coding of what it calls "symbols" based on a multi-level arithmetic coding scheme. Petition at 10. A POSITA would be aware

that multi-level arithmetic coding is distinct and different from binary arithmetic coding.

55. Murashita explains that what it calls "symbols" are "events of characters," which are symbol values. Ex-1007 at 1:63-2:4. Murashita's Figure 1a further confirms this, showing the frequency (probability) of certain symbol values a, b, c, d and e occurring and equating each to a fractional range. Murashita provides further examples that confirm this, such as the example of encoding a character string "abe" (Ex 1007 at 2:5-20) where each of the symbol values "a", then "b", then "e" are described as "symbols". *See also* Ex 1007 at 3:11-24.

56. Murashita focuses on the process of context updating, specifically how many times a context should be updated. Murashita discloses a "controlling unit 13 [that] controls the context holding/updating unit 12 to update the context data. In doing so, the controlling unit compares a predetermined threshold value with a cumulative frequency of occurrence for all the symbols registered under a given context. Then, only when the cumulative frequency is smaller than the threshold value, the controlling unit 13 controls the context holding/updating unit 12 to update the context data for the given context." Ex-1007 at 6:40-47. In other words, Murashita suggests pausing the updating of context data when certain conditions are met. The "symbols" considered in Murashita are the symbol values used in entropy coding. This is clear from Murashita' s description of the symbols

and Figures 1A and 1C. For example, symbol values a through e represent different values of a text character and have a unique frequency and range associated with each. The accompanying description in Murashita discusses the use of a string of symbol values in arithmetic coding. *See* Ex-1007 at 2:5-20 (describing an example of encoding a character string "abe" where each of the symbol values "a", then "b", then "e" are described as "symbols"); *See also* Ex-1007 at 3:11-24.

57. Each time a symbol value (such as the value of a text character) registered to a specific context is coded, that context is updated with the new frequency and may reorder the context accordingly. Ex-1007 at 3:57-4:15; 7:24-39. Murashita performs this process for each symbol registered to a context and tracks the cumulative frequency of all symbol values registered to the context. When the cumulative frequency exceeds a certain threshold, the process of updating the context is no longer performed. Ex-1007 11:25-12:12.

58. Murashita considers the number of times a context tree has been updated and ceases the context updating process after coding a certain number of symbol values. Murashita does not discuss syntax elements, a term never mentioned in Murashita. Further, Murashita does not disclose or discuss a frequency of occurrence of syntax elements. Further, Murashita does not discuss bypassing context updating and the word "bypass" does not appear in Murashita.

b. Marpe (Ex-1006)

59. U.S. Patent Application Publication No. 2005/0253740 ("Marpe"), titled "Method and Apparatus for Binarization and Arithmetic Coding of a Data Value" was filed on May 3, 2005. Petitioner explains that Marpe's approach is "based on binarization and binary arithmetic entropy coding". Binary arithmetic coding (BAC) is a distinct and different approach from multi-level arithmetic coding. Marpe discusses a breakdown of syntax element or symbol values using the concept of bins. A bin is a binary value of 0 or 1 that represents the smallest addressable unit in entropy coding. A series of bins (a bin string) can represent a single, non-binary symbol value, one or more of which may be used to represent a syntax element in the bitstream. Marpe's disclosure of "the most frequently observed bins" refers to bins that correspond to smaller absolute values, which would typically occur first in the order of binarization of a non-binary value. Thus, in Marpe, the frequency of observation of bins does not relate to the frequency of occurrence of syntax elements but instead relates to the frequency of occurrence of syntax element values. For example, consider Marpe Fig. 5, which illustrates a set of binarizations of a syntax element with a range of possible absolute values ('Abs. value'). Bin 1 is present in all of the bin strings (binarizations). Bin 2 is present for all absolute values greater than 1, and so on. Thus bin 1 is the most frequently observed, followed by bin 1 and so on. The bin strings are specified with the

assumption that small absolute values are more frequently observed than large absolute values. Thus, the frequency of observation of a particular bin (such as bin 2) relates to the expected frequency of occurrence of syntax element values, in this case the expected frequency of occurrence of absolute values of the syntax element in question.

c. Yu (Ex-1018)

60. Yu describes a hardware-oriented approach for implementing CABAC decoding in H.264/AVC systems. The H.264 CABAC engine was a known prior art method referenced in the specification of the '701 Patent. *See* '701 Patent at 4:23-42. Yu never proposes changing H.264's CABAC, only implementing it in what Yu describes as an efficient way. Yu explains that CABAC significantly increases computational complexity compared with H.264's CAVLC entropy coding technique, because H.264 CABAC encodes video information as a sequence of binary symbols—called "bins"—each representing a 0 or 1, and an H.264 decoder must maintain and continually update 399 probability models that estimate whether each upcoming bin is more likely to be a 0 or a 1. Ex-1018 at 1352–53. Yu identifies two operations as dominating CABAC decoding: (1) the arithmetic decoding of these binary symbols, and (2) the loading, updating, and writing back of the associated context (probability) models for each symbol. *Id.* Yu acknowledges that, when performed in software, "about 4500 cycles are needed" merely for decoding

the binary symbols of a single macroblock. *Id.* To address these limitations, Yu presents a hardware structure organized around two decoding trees—one for regular binary symbols and one for bypass-mode symbols. *Id.* Yu states that its hardware attempts to decode two regular binary symbols ("bins") per cycle and, for specific syntax elements that contain a unary prefix followed by a sign bit encoded in bypass mode, the hardware attempts to decode that bypass sign bit "within the same cycle of the last regular bin of the prefix." *Id.* at 1355.

61. For binary arithmetic decoding itself, Yu explains that the decoder maintains an interval called *range* and subdivides it into two probability-based subintervals: rLPS, which corresponds to the estimated probability interval for the "least-probable symbol," and rMPS, which corresponds to the estimated probability interval for the "most-probable symbol." *Id.* at 1353. The decoder compares each incoming *offset* value to these subintervals (rLPS, rMPS) to determine whether the next symbol (i.e. the next decoded bin) should be decoded as the more-probable symbol or the less-probable symbol, and then decodes the bin as 0 or 1 and updates the *range* and *offset* registers accordingly. *Id.* Yu divides H.264 CABAC's 399 probability models into 18 groups that are loaded from memory and written back group-by-group rather than on a per-symbol basis. *Id.* at 1356, 1358.

62. The Petition makes several incorrect statements about Yu's disclosure. I will deal with these when discussing the Grounds in the Petition.

VIII. THE GROUNDS IN THE PETITION DO NOT RENDER OBVIOUS THE CHALLENGED CLAIMS

63. Each of Petitioner's grounds fail to disclose or render obvious the Challenged Claims of the '701 Patent at least because all fail to disclose or teach measuring or using a frequency of occurrence of syntax elements. Instead, all references consider characteristics of symbol values in determining any changes to the entropy coding methods used.

1. Petitioner Has Not Established That The Challenged Claims Are Unpatentable In Light Of Ground 1—Murashita.

a. Petitioner fails to demonstrate that limitations 1[a], 8[a], and 15[a] are obvious over Murashita

64. Each of limitations 1[a], 8[a], and 15[a] are written with similar language and require "determining a frequency of occurrence threshold based on an expected frequency of occurrence of syntax elements in a bit stream." In my opinion, Petitioner[1] has failed to show that Murashita teaches or suggests this limitation.

65. Petitioner equates "data symbols" with the claimed "syntax elements."

[1] Petitioner put forth an expert declaration that in my view is substantively similar to the limitation-by-limitation analysis. Therefore, references to "Petitioner" as used in these claim sections refers to both Petitioner's argument and the substantively similar expert declaration substance.

According to Petitioner, syntax elements are simply elements of data represented in the bitstream and therefore constitute data symbols. Petition at 13.

i. Murashita's Data Symbols Are Not the Claimed "Syntax Elements"

66. Murashita does not teach or even discuss the "frequency of occurrence of syntax elements" as required by the challenged claims. Instead, Murashita is directed to an adaptive arithmetic coder that tracks only the frequency of occurrence of *symbols*—a different concept. Each of the symbols disclosed in Murashita is a symbol value (i.e. one of many possibilities a symbol may take). *See supra* § VII.2.a. For example, Murashita explains that "frequency of occurrence is counted for each symbol" (Ex-1007 at 3:21-24) and repeatedly emphasizes that the system maintains a dictionary in which "frequency and cumulative frequency are assigned to each symbol" (Ex-1007 at 3:66-67). Notably, Murashita describes a device that "codes the symbols based on frequency of occurrence of the symbols," not syntax elements. *Id.* at Abstract.

67. Murashita does not disclose or suggest syntax elements or any mapping from syntax elements to symbols. To the contrary, Murashita's update process is defined purely in terms of symbol statistics: the system "increments the frequency of the symbol K by 1" and "increments the cumulative frequency of all symbols listed above the symbol K." Ex-1007 at 8:24-28. Thus, Murashita is focused

only on symbol-level frequency counting, i.e. counting the frequency of occurrence of particular symbol values, and does not teach or imply frequency analysis of syntax elements, nor does it teach or imply measuring or responding to the frequency of occurrence of syntax elements.

68. It is my opinion that a POSITA would understand that a syntax element is an element of data represented in the bitstream, whereas a symbol is an element used to represent syntax elements—such as a value, a decision, or a bin. This distinction is confirmed by Petitioner's own prior art. Marpe expressly states that "[e]ach syntax element is a data value having a certain meaning in the video signal bit stream that is passed to the entropy coder." Ex. 1006 ¶38. Likewise, Yu explains that "the syntax element is an element of data represented in the bit stream to be encoded," and that syntax elements are mapped to bin strings, where "each bin of the bin string is arithmetically encoded into the bit stream." Ex. 1018 at 1353. Yu further clarifies that a "bin is a binary value." *Id.* at 1352. Thus, syntax elements are meaningful constructs of the bitstream, while symbols (including bins) are the unit of information passed to or from the entropy coder. The frequency of occurrence of a symbol value (e.g. how often does a character take the values 'a', 'b', 'e', etc.) is not the same as the frequency of occurrence of a syntax element (e.g. how often does a particular syntax element occur). Murashita addresses only symbol values—not syntax elements.

69. In video coding, a single syntax element may be mapped into multiple symbols, or a symbol may represent only a portion of a syntax element. Thus, while syntax elements are typically coded as symbols, the two concepts are not interchangeable, and a syntax element does not become a "symbol" merely by virtue of being entropy coded.

70. Thus, Murashita does not render independent claims 1, 8, or 15 obvious because Murashita operates exclusively on symbol values, not syntax elements, and does not operate on or even consider the frequency of occurrence of syntax elements. Because dependent claims 6, 13, and 20 incorporate the limitations of their respective independent claims, Murashita likewise cannot render those dependent claims obvious.

ii. Murashita Does Not Teach The '701 Patent's Frequency-Based Categorization of Syntax Elements and Its Dual-Mode Entropy-Coding Behavior

71. It is my opinion that the '701 Patent fundamentally differs from Murashita because the '701 Patent expressly requires categorizing syntax elements based on their frequency of occurrence and then applying different entropy-coding modes depending on the assigned category, whereas Murashita teaches neither concept. The '701 Patent specifies that a coding apparatus "determin[es] a frequency of occurrence threshold" and "categorize[s] a plurality of syntax elements … based on the frequency of occurrence of the syntax elements." '701 Patent at 14:5–16. The

’701 Patent then uses this categorization to select between the use of two distinct entropy coding behaviors: (1) context-updated entropy coding symbols corresponding to the first category of syntax elements and (2) bypass-mode entropy coding—using a fixed 50% probability—for symbols corresponding to the second category of syntax elements. *Id.* at 10:7–39.

72. Murashita, in contrast, does not compute or use any frequency-of-occurrence statistic for syntax elements, does not categorize syntax elements, and does not provide any dual-mode entropy-coding framework. Murashita exclusively updates and reorders symbol value frequency tables used in multi-level arithmetic coding, with no disclosure of syntax elements or category-specific entropy-coding operations. For example, Murashita recites “At a step S101, ***a symbol K*** is coded by the coding unit 102. At a step S102, a check is made by the comparison unit 109 whether the cumulative frequency is less than the threshold value. If it is, the procedure proceeds to a step S103. If it is not, the procedure ends… At a step S103, the sorting unit 105 searches for ***a symbol N*** which is located at the top of the list among symbols having the same frequency as that of the ***symbol K***. At a step S104, the ***symbol K*** is exchanged with the symbol N. At a step S105, the frequency updating unit 107 ***increments the frequency of the symbol K*** by 1.” Ex-1007 at 8:34-39, 8:49-54, *emphasis added*. Thus, the ’701 Patent’s core syntax element frequency-based categorization and corresponding context-based or bypass

processing are entirely absent from Murashita.

iii. Murashita Does Not Teach Bypassed Context Updating

73. The Petition incorrectly asserts that Murashita's cessation of its context-update procedure is equivalent to "bypassed context updating." Petition at 43-44. That is not the case. Murashita stores "context data" indicating frequency of occurrence, cumulative frequency of occurrence and an order. Ex-1007 at 6:32-35. As shown in Murashita's Figure 10, there are only two operational possibilities when a symbol is received: either (1) continue performing the update operations on the symbol-frequency data or (2) discontinue further updates when a threshold is reached. Murashita's "discontinue updating" state simply halts its symbol-frequency maintenance; it does not initiate a second, alternative coding mode.

74. By contrast, independent claims 1, 8, and 15 of the '701 Patent expressly require two distinct treatments of symbols during entropy-coding 1) symbols that have been subjected to a context updated; and 2) symbols that have bypassed context updating. Simply ceasing the context updating for symbols does not constitute bypassing context updating because the symbols have already been subject to a context updated in previous cycles. Murashita's single-mode arithmetic coder always codes symbols using context data that has been updated, i.e. always entropy codes symbols that have been subjected to a context update, using stored context data. Murashita does not disclose or contemplate a "bypass mode" in which

symbols are coded without the use of context data and/or with bypassing of context updating. Murashita therefore cannot satisfy, either individually or in combination, the claim-required dual-mode entropy-coding framework.

iv. Murashita Does Not Disclose Categorizing Syntax Elements Based on the Frequency of Occurrence of Syntax Elements.

75. As discussed above, Murashita discusses considering characteristics of symbol values, not syntax elements. *See Supra* § VII.2.a. However, even if Murashita did disclose categorizing syntax elements (which it does not), it does not disclose categorizing based on the frequency of occurrence of syntax elements or symbol values.

76. Murshita's cumulative frequency does not "represent the expected individual frequency of occurrence of a symbol" as Petitioner suggests. Petition at 28. For example, Murashita's Fig. 4A provides exemplary frequencies and cumulative frequencies.

FIG. 4A

ORDER OF FREQUENCY	SYMBOL	FREQUENCY	CUMULATIVE FREQUENCY
0	-----	00	327
1	00	10	317
2	01	7	310
3	02	4	306
⋮	⋮	⋮	⋮
n − 1	'M'	2	m2
n	'N'	1	m1
⋮	⋮	⋮	⋮
X	'A'	1	m0
⋮	⋮	⋮	⋮
254	FD	1	3
255	FE	1	2
256	FF	1	1
257	EOF	1	0

77. In this example, symbol 01 has a frequency of occurrence of 7 and a cumulative frequency of 310. Further, symbols FD and FF have identical frequencies of occurrence (1) but different cumulative frequencies (3 and 1). The cumulative frequency provides no information as to the "individual frequency of occurrence" of a symbol. Indeed, symbol 01 has a frequency of occurrence seven times larger than the frequency of occurrence of symbol FF and a cumulative frequency of occurrence 310 times larger than the cumulative frequency of symbol FF. As such, Murashita's cumulative frequency of occurrence does not even correlate with the frequency of occurrence of a symbol.

2. Petitioner Has Not Established That The Challenged Claims Are Unpatentable In Light Of Ground 2—Murashita In Combination With Marpe.

78. As I explain in detail below, there are multiple reasons why Murashita does not teach or suggest to a POSITA the limitations of independent claims 1, 8, or 15.

a. Petitioner fails to demonstrate that limitations 1[a], 8[a], and 15[a] are obvious over Murashita

79. Each of limitations 1[a], 8[a], and 15[a] are written with similar language and require "determining a frequency of occurrence of syntax elements in a bit stream." '701 Patent, 14:7-9. In my opinion, Petitioner has failed to show that Murashita teaches or suggests these limitations for the reasons discussed above.

b. Marpe Does Not Categorize Syntax Elements Based on "Frequency of Occurrence"

80. Petitioner's reliance on Marpe fails because Marpe nowhere teaches, suggests, or contemplates categorizing syntax elements based on their "frequency of occurrence," let alone determining a frequency threshold that governs entropy-coding operations. This limitation is not disclosed or suggested in Marpe or indeed any of the cited prior art.

81. The '701 Patent expressly requires that syntax elements be classified according to how often they occur. For example, claim 1 recites determining "a frequency of occurrence threshold based on an expected frequency of occurrence of syntax elements in a bit stream" and "categorizing a plurality of syntax elements… wherein syntax elements which occur greater than the frequency of occurrence threshold are categorized into the first category and syntax elements which occur less than the frequency of occurrence threshold are categorized into the second category."

82. The '701 Patent then requires applying different entropy-coding treatments to each category—coding with context updating for one category, coding with bypassed context updating for the other.

83. Marpe contains no such teaching. Marpe explains only that in CABAC, a binary decision enters "the context modeling stage, where a probability

model is selected such that the corresponding choice may depend on previously encoded syntax elements or bins." Ex-1007 at ¶ 0119. Marpe further states that in bypass mode, "the bypass coding engine… works preferably with a fixed probability model without any context adaption." *Id.* at ¶ 0120. These passages confirm that the distinction between regular and bypass modes in Marpe is predetermined by syntax-element type and bin characteristics—not by a "frequency of occurrence threshold." Marpe never states, implies, or suggests that the choice of coding path depends on how often any syntax element appears in the bitstream.

84. Equally important, Marpe's context-modeling mechanisms are structurally based, not frequency-based. Marpe states that "[f]our basic design types of context models can be distinguished," including models that use "up to two neighboring syntax elements," models where "values of prior coded bins" are used, and models that depend on block type or scanning position. *Id.* at ¶¶ 0130–0132.

85. None of these mechanisms involve determining an expected frequency of occurrence, computing a threshold, or dividing syntax elements based on whether they occur more or less frequently than such a threshold. Marpe's modeling is focused on local context and bin position, not statistical occurrence frequency.

86. Marpe further confirms that bypass mode is selected according to bin type—not based on any syntax element occurrence statistics. Marpe explains that bypass mode "is especially effective when coding the bins of the primary suffix of

those syntax elements concerning components of differences of motion vectors and transform coefficient levels." *Id.* at ¶0120. This is a fixed rule based on the symbol suffix of the binarization of syntax-elements. There is no suggestion that bypass mode is chosen because certain syntax elements occur rarely or frequently.

87. Thus, Marpe neither discloses nor suggests the claimed "frequency of occurrence threshold" or the required categorization of syntax elements into two groups based on their expected frequency of occurrence. Marpe's CABAC framework is structurally driven, not statistically driven.

88. Because the '701 Patent's frequency-based categorization is absent from Marpe, Petitioner cannot meet this limitation under obviousness.

c. Murashita in view of Marpe Cannot Be Combined Because They Would Produce an Unworkable Result

89. The Petitioner contends that "[a] POSITA would have been motivated to build on Murashita's efficiency gains by swapping out its "multi-level arithmetic coding" engine for Marpe's very well-known and more efficient CABAC engine, which is based on a binary arithmetic coding scheme rather than multilevel. Petition at 54.

90. Even if a POSITA tried to combine Murashita with Marpe, the result would be technically unworkable. The two systems rely on fundamentally incompatible probability-modeling architectures and update mechanisms.

Attempting to merge them would break essential functionality in both and yield a system that could not operate.

91. Murashita is entirely directed to multi-level arithmetic coding and built around a single, global symbol-frequency table that governs probability assignments for all encoded symbols. As Murashita explains, its coder "[i]n the multi-level arithmetic coding method described above, ranges are divided according to frequency of occurrence. Thus, *a dictionary* [global table] as shown in FIG. 4A is necessary for counting and recording an occurrence of each symbol." Ex-1007, col. 3:57–60. This global table, as illustrated in Figure 4A, is the core of Murashita's approach: it sets updated cumulative frequencies, determines symbol intervals, and controls how arithmetic subdivision proceeds. Murashita's arithmetic encoding is based on the principle of subdividing a range into subranges, one for each possible symbol value (e.g. Murashita Fig. 1A, showing 5 symbol values and 5 subranges).

92. Marpe, by contrast, does not use a dictionary or a global frequency table, and instead depends on a Context-based Adaptive Binary Arithmetic Coding (CABAC) architecture. Every syntax element is first binarized into a sequence of binary "bins," each of which is independently assigned a context model based on neighboring syntax elements, prior bins, block type, or scanning position. Marpe explains that in such a scheme, binary arithmetic coding is used instead of multi-level ("m-ary") arithmetic coding (Marpe at 0014). Marpe explains that in CABAC,

"a given non-binary valued syntax element is uniquely mapped to a binary sequence" and that in regular mode "a probability model is selected" based on previously encoded bins. Ex-1006 at ¶¶ 0115, 0119. Unlike Murashita, Marpe's probability models are independent across contexts and are updated adaptively.

93. Because the two systems rely on incompatible update mechanisms and incompatible arithmetic coding techniques, I fail to see how they could be combined, even though I have studied and written about both m-ary and binary arithmetic coding for over 20 years. Murashita's algorithm requires updating one global symbol frequency table (Murashita's dictionary as illustrated in FIG. 4A) after every symbol, whereas Marpe's CABAC encoder is built entirely around multiple local contexts whose states diverge based on the sequence of entropy coding decisions made when encoding a specific video sequence.

94. Conversely, inserting Marpe's CABAC context-model updates into Murashita's global frequency table would not be compatible with the cumulative-frequency structure needed for arithmetic interval subdivision, rendering Murashita's coder inoperable.

95. Accordingly, the combination of Murashita and Marpe would not yield a workable entropy-coding system, and Petitioner's theory fails as a matter of technical feasibility.

96. Further, Murashita is directed to general purpose compression of e.g.

text files (Ex-1007 at 1:7-13, 1:33-37), whereas Marpe is directed to the specific application of video compression (Ex-1006 at ¶ 3). A POSITA would be aware that multi-level (m-ary) arithmetic coding can more accurately model the probability of occurrence of multiple symbol values than binary arithmetic coding (BAC), in which a particular symbol value must be approximated as a set of binary nodes. Ex-1006 at ¶ 16. Marpe's rationale for using BAC is the finding that it can offer "effective compression… in particular of … motion vector differences or transform coefficient level values." Ex-1006 at ¶ 14. Such a rationale would not apply to Murashita's general purpose compression algorithm.

97. Murashita's goal is to solve a problem identified in compression of text and similar files, which can lead to "a long processing time." Ex-1007 at 5:4-10. Murashita purports to solve this problem by selectively updating its cumulative frequency table. A POSITA would not be motivated to attempt a different (and incompatible) solution of "swapping out" Murashita's m-ary arithmetic coder for Marpe's CABAC engine. Petition at 54.

98. Petitioner's proposed architecture has no technical foundation but is merely a "cut and paste" of diagrams from Marpe and Murashita. Petition at 57. In Marpe Fig 2, the decision between regular and bypass modes depends on the binarization and the syntax element value. *See e.g.*, Ex. 1006 at ¶¶ 81-83. Petitioner suggests that the switch between context-based or bypass coding **per bin** in Marpe's

system should be controlled instead by Murashita's cumulative frequency of **symbol values** and that somehow Murashita's system would "become more efficient" as a result. Petition at 58. It is unclear how this would work and what effect the control of bin coding based on cumulative frequency of symbol values would have on compression efficiency. Petitioner is silent on these points. Again, as someone with more than 20 years of expertise in m-ary and binary arithmetic coding systems, I am at a loss to know the answer. Nothing in Murashita, Marpe or the Petition provides any rationale or motivation for such a modification. Petitioner's arguments appear to be nothing more than hindsight using the '701 Patent as a roadmap.

d. No Reasonable Expectation of Success in Combining Murashita with Marpe

99. For the reasons described in § VIII.2.c., a POSITA would not have been motivated to modify Murashita's system/process in view Marpe in the manner proposed by the Petition, nor would a POSITA have had a reasonable expectation of success regarding the outcome.

100. Even if a POSITA were to successfully combine Murashita with Marpe, the combination would not satisfy the Challenged Claims at least because neither reference discloses categorizing syntax elements based on their frequency of occurrence, nor entropy coding symbols based on such a categorization, as I have explained above.

3. Petitioner Has Not Established That The Challenged Claims Are Unpatentable In Light Of Ground 3—Yu

101. Yu discloses a prior art entropy coding architecture, i.e. an implementation of H.264's CABAC without changing CABAC at all. The disclosures in Yu relate not to an improvement in the entropy coding process but rather a purported efficient hardware implementation.

a. Yu's Syntax Element Categories Do Not Determine Whether They Are Context-coded or Bypass-coded

102. Yu divides syntax elements into two classes *based on bin-length*, not based on whether the bins are regular-coded ("context" coded) or bypass-coded. Yu explains that "there are two classes of syntax elements": a first class that "contain a few bins and occur only once or a few times in one MB [macroblock]," and a second class that "contain several bins and occur tens or hundreds of times in one MB [macroblock]." Ex-1018 at 1355. For example, the first group has "***one bin*** [that] is processed per cycle" and the second group has ***three bins***, "two regular [context updated] bins together with one bypass bin" processed per cycle. *Id*, *emphasis added.* In Yu, there is no distinction as to whether the first group's bin is bypass or regular ("context") coded.

b. Yu's syntax element categories are unrelated to the comparison of *offset* to an alleged threshold.

103. Petitioner makes a technically incorrect statement, attempting to link Yu's two "classes" with the determination of LPS or MPS. *See* Ex-1018 at 1355. It

is simply untrue that syntax elements that contain a few bins and occur only once or a few times in one MB, correspond or relate to "bins where the offset value is below the threshold, indicating an LPS." Petition at 89. It is untrue that syntax elements that contain several bins and occur tens or hundreds of times in one MB, correspond or relate to "bins where the offset value is above the threshold, indicating an MPS." Petition at 89. The whole point of binary arithmetic coding is that **every** bin coded in "regular mode" has the possibility of being an LPS (least probable symbol value, 0 or 1) or an MPS (most probably symbol value, 0 or 1). Yu explains this clearly, Ex-1018 at 1353-1354. Thus contrary to Petitioner's technically incorrect statement (Petition at 89), there is no link between Yu's first and second "classes" and the comparison of *offset* to the subintervals of rLPS and rMPS.

c. Yu's Use of Bypass Is Not Related to the "Frequency of Occurrence of Syntax Elements"

104. Yu does not suggest that bypass coding is triggered by, or in any way related to, the "frequency of occurrence of syntax elements." In fact, Yu explains that the decision to use bypass mode is dictated solely by the CABAC coding rules for specific syntax-element structures, not by how frequently a syntax element is observed to appear. Yu states that "bins encoded by bypass mode only appear in two cases, either as sign or as Exp-Golomb (EG) [suffix] code," and further notes that the EG suffix occurs only when certain values (e.g., abs_mvd greater than 8 or

coeff_abs_level_minus1 greater than 13) are present, circumstances Yu characterizes as "rare occasions" because "at least 90%" of these syntax elements have small values that do not trigger bypass suffixes. Ex-1018 at 1355. Thus, in Yu (as in H.264), bypass coding is determined by whether a value is represented as a sign bit or EG suffix and not by the rate or frequency with which a syntax element arises. Yu's disclosure contradicts Petitioner's incorrect statement that in Yu, "high-occurrence [syntax] elements are decoded in bypass mode, where context is not needed." Petition at 84. On the contrary, in Yu (as in H.264), bypass coding is applied to bins that have been determined to have approximately 50% probability of being 0 or 1, namely (i) sign, which is approximately equally likely to be positive or negative, and (ii) the aforementioned infrequently-occurring Exp-Golomb suffix bins.

105. Yu's discussion of frequent versus infrequent syntax elements appears only in the context of hardware scheduling—i.e., whether a syntax element is placed in the "first class" (processed one bin per cycle) or "second class" (processed two regular bins, and when applicable, one bypass sign bit per cycle)—and not as a basis for selecting bypass versus context-coded arithmetic decoding. Ex-1018 at 1355. Nothing in Yu links bypass coding to frequency of occurrence; rather, bypass mode is strictly determined by the internal coding structure of the syntax element, not how often that element appears. Further, Yu's discussion of the "first class" and "second

class" of syntax elements does not in any way relate to categorizing syntax elements based on a comparison with a frequency of occurrence threshold as required by the claims of the '701 Patent. Instead, Yu's "first class" and "second class" are categories of syntax elements that have already been defined in the H.264 Standard, e.g. "Those syntax elements in the second class include abs_mvd, significant_coeff_flag, last_significant_coeff_flag and coeff_abs_level_minusl." Ex-1018 at 1355.

d. Yu's Probability Monitoring Is Not "Expected Frequency of Occurrence of Syntax Elements"

106. Petitioner contends that Yu's CABAC framework expressly teaches the claims' frequency-based categorization and threshold by modeling probabilities for bins and splitting range into rLPS and rMPS such that, in "regular mode," "the LPS probability pLPS is estimated by the corresponding context model," while in "bypass mode," "pLPS is estimated as 0.5 and no context model is needed," thereby establishing a threshold between LPS and MPS decisions based on expected occurrence frequency. *See* Petition at 82–84, 88–90.

107. According to the Petition, Yu "classifies SEs [syntax elements] into two classes based on frequency of occurrence: a first class with SEs that 'occur only once or a few times in one MB [macroblock],' and a second class with SEs that 'occur tens or hundreds of times in one MB [macroblock],'" and uses a fixed

decision rule—"range/2 to determine *the bin* is MPS or LPS"—in bypass mode, reinforcing that the categorization turns on expected frequency. *Id.* at 83–84, 88–89. The Petition further argues that "the decision point between decoding *a bin* as LPS or MPS (the threshold) thus indicates the frequency of occurrence of symbols at which context updating occurs," because regular-mode decoding employs context-updated probability estimation whereas bypass mode omits it for high-frequency elements. *Id.* at 90.

108. ***First***, Yu processes and determines probabilities one bin at a time, not for any batch or set of bins which make up a syntax element. Yu explains that a "bin is a binary value and the basic arithmetic decoding unit in CABAC" meaning each bin is just a single binary symbol (0 or 1). Ex-1018 at 1352.

109. It is my opinion that Yu repeatedly makes clear that the probability lookup, the decision of LPS versus MPS, and the context model occur per bin, not per group of bins—e.g., "[b]efore arithmetically encoding one bin, the corresponding context model is selected…and the model is updated after the bin is encoded." *Id.* at 1352. Thus, even when Yu's hardware structure attempts to decode more than one bin in each cycle for speed, the probability estimation and decision logic described in Yu remain fundamentally bin-by-bin operations, each relying on the model's probability state for that specific bin.

110. ***Second***, Petitioner's reliance on Yu's "probability modeling" to

suggest disclosure of an "expected frequency of occurrence of syntax elements in a bitstream" is misplaced. Yu's probability modeling concerns only the likelihood that a decoded bin (i.e., a single binary symbol) is a 0 or 1 and is not tied to how often any syntax element appears in the bitstream. Yu expressly defines probability modeling as the process by which "each context model… corresponds to a certain bin," and where each model stores (1) "a 6-bit state value… as the probability index" and (2) "a binary value of the most probable symbol (MPS)." Ex. 1018 at 1353. The purpose of the model is to estimate *pLPS*, "the estimated probability value of the least probable symbol", and then subdivide the arithmetic-coding interval into *rLPS* and *rMPS* based on that per-bin probability. *Id.*

111. Yu does not tie this probability modeling to how frequently a syntax element appears in the video stream. Indeed, Yu makes clear that syntax elements are separate constructs that "are uniquely mapped to a bin string" consisting of individual bins, and the probability model operates at the level of each bin within that string—not at the level of the syntax element itself. *Id.* at 1353.

112. Accordingly, Yu's probability modeling relates only to symbol-level probability estimation (i.e. the probability that a bin is 0 or 1), not any expected rate or frequency at which a syntax element appears in the encoded bitstream. Further, even when Yu decodes "two regular bins together with one bypass bin in one cycle" (Ex. 1018 at 1355), the hardware is not applying one probability model

to all three bins. Instead, each regular (i.e. context coded) bin uses its *own* context state and its *own* rLPS/rMPS values. Bypass bins, by definition, do not use a probability model at all and rely on a fixed assumption of a probability of 0 or 1 being 0.5. *Id.* at 1353–54.

e. Petitioner's Reliance on Yu's Threshold is Misplaced

113. Petitioner's reliance on Yu's "threshold" is misplaced. *See* Ex-3001 at 90. The alleged "threshold" Yu describes is not a measure of how frequently a syntax element appears in a bitstream, nor is it any kind of occurrence-based criterion.

114. Instead, Yu expressly explains that the arithmetic decoder divides the current interval into two probability-based subintervals—rLPS and rMPS—where "rLPS correspond[s] to the estimated probability interval of the LPS" and rMPS correspond[s] to the interval of the MPS. Ex. 1018 at 1353. The decoder determines whether a bin is an MPS or LPS solely by checking "whether the offset falls into the subinterval of rLPS or rMPS." *Id.* This comparison is the alleged "threshold" Petitioner references, but it is nothing more than a probability-model decision applied to a single bin value (i.e., whether that bin is more likely a 0 or 1). It has no connection to how often a syntax element has appeared in the bitstream and plays no role in determining the coding mode (context-based or bypass) of that bin. This alleged threshold has no relationship with Yu's "first class" and "second class" of less frequent or more frequent syntax elements. Ex 1018 at 1355.

115. In fact, Yu's alleged threshold (rMPS) is only applied to bins that the encoder/decoder has already determined are to be coded using a context update. In the event that the encoder/decoder is performing bypass coding, a static 50/50 probability is used and there is no LPS or MPS, as they are equiprobable. Bypass bins therefore do not make any use of the rMPS and instead the "*offset* value is compared with *range* to determine [sic] the bin is 1 or 0." Ex-1018 at 1354.

116. Thus even if rMPS were the claimed threshold, which it is not, in Yu rMPS is not used in any way to categorize syntax elements (or bins for that matter) into two categories such that one category is coded using bypass mode.

117. In my opinion, it is not enough to point to an arbitrary alleged threshold. The Challenged Claims of the '701 Patent require that the threshold is used to categorize syntax elements and to code symbols corresponding to the syntax elements of each category using context updating or bypassing context updating based on the categorization.

118. Thus, it is my opinion, for the reasons discussed above, Yu does not teach independent claims 1, 8, and 15.

119. Claims 3, 6-7, 10, 13-14, 17, 20 depend on the independent claims. For the reasons discussed above, Petitioner has failed to show that Yu discloses these claims.

f. As to Dependent Claims 6, 13, and 20, Yu's Threshold is Not "Predefined"

120. It is my opinion that Yu does not disclose any "predefined" or fixed "threshold" for determining a bin value. Instead, Yu explains that the decision boundary used during arithmetic decoding is dynamically generated from the probability model stored in the context state, not from any preset threshold. Ex-1018 at 1353. As Yu states, the decoder subdivides the coding interval into two subintervals "rLPS corresponding to the estimated probability interval of the LPS" and "rMPS, which is assigned to the MPS" *Id.*

121. The decoder then determines the bin value by checking "whether the offset falls into the subinterval of rLPS or rMPS." *Id*. Both rLPS and rMPS are obtained from a lookup table indexed by the *current state* of the context model—"a 6-bit state value… used as an index of the estimated probability value." *Id.*

122. Yu further confirms that the probability state is "updated after encoding each bin," meaning that the threshold changes from bin to bin because it depends on the continuously updated probability estimate. Thus, Yu's decision boundary is neither fixed nor "predefined"; it is a function of the evolving probability model for each bin and cannot operate as a predetermined or "predefined threshold."

123. Thus, for the reasons discussed above, Yu does not teach dependent claims 6, 13, and 20.

IX. SIGNATURE

This Declaration represents the opinions that I have formed up to date based on the material currently available to me. I reserve the right to revise, supplement, and/or amend my opinions stated in this Declaration based on new information that becomes available to me and on my continuing analysis of the materials available to me.

I declare that all statements made herein of my knowledge are true, and that all statements made on information and belief are believed to be true, and that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine, imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Dated: November 28, 2025

Dr. Iain Richardson