

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AMAZON.COM, INC. and AMAZON.COM SERVICES LLC,
Petitioners

v.

NOKIA TECHNOLOGIES OY,
Patent Owner.

IPR2024-00572

U.S. Patent No. 7,724,818

**PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 7,724,818**

Mail Stop PATENT BOARD
Patent Trial and Appeal Board
U.S. Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	STANDING	1
III.	IDENTIFICATION OF CHALLENGE	1
A.	Prior Art	1
B.	Grounds	2
IV.	OVERVIEW	2
A.	Background of Technology and Admitted Prior Art	2
B.	Purported Invention.....	6
C.	Prosecution.....	7
V.	LEVEL OF ORDINARY SKILL	9
VI.	CLAIM CONSTRUCTION	9
A.	Means-Plus-Function (MPF).....	9
VII.	DETAILED EXPLANATION OF GROUNDS.....	11
A.	Ground 1: Obviousness over JVT-E022d7	11
1.	JVT-E022d7	11
2.	Claim 6.....	13
3.	Claim 7	23
4.	Claim 8	24
5.	Claim 9.....	25
6.	Claim 11	25
7.	Claim 15	27
8.	Claim 21	29
9.	Claim 23	30
B.	Ground 2: Obviousness over Linzer	33
1.	Linzer	33
2.	Claims 6-9, 11, 15, 21, and 23	34
C.	Ground 3: Obviousness over Au	34

TABLE OF CONTENTS (continued)

	Page
1. Au.....	34
2. Claim 6.....	37
3. Claim 7.....	43
4. Claim 8.....	44
5. Claim 9.....	45
6. Claim 11.....	46
7. Claim 15.....	49
8. Claim 21.....	50
9. Claim 23.....	52
D. Ground 4: Obviousness over VCEG-N52 in view of Sriram	55
1. VCEG-N52	55
2. Sriram.....	57
3. Motivation to Combine VCEG-N52 and Sriram	59
4. Claim 6.....	61
5. Claim 7.....	70
6. Claim 8.....	70
7. Claim 9.....	72
8. Claim 11.....	72
9. Claim 15.....	75
10. Claim 21.....	76
11. Claim 23.....	77
VIII. DISCRETIONARY FACTORS FAVOR INSTITUTION	82
A. <i>Fintiv</i>	82
B. 325(d).....	82
IX. CONCLUSION.....	85
X. MANDATORY NOTICES	85

TABLE OF CONTENTS
(continued)

	Page
A. Real Party in Interest.....	85
B. Related Matters	85
C. Notice of Counsel and Service Information.....	86

EXHIBIT LIST

Exhibit	Reference
1001	U.S. Patent No. 7,724,818 (“the ’818 patent” or “’818Pat.”)
1002	Certified File History of the ’818 patent (“’818FH”)
1003	Declaration of Dr. Eli Saber (“Saber”)
1004	Curriculum Vitae of Dr. Eli Saber
1005	U.S. Patent No. 7,660,356 to Linzer (“Linzer”)
1006	Certified File History of Linzer (“LinzerFH”)
1007	Editor's Proposed Draft Text Modifications for Joint Video Specification (IUT-T Rec. H.264 ISO/IEC 14496-10 AVC), Draft 7 (“JVT-E022d7”) (excerpted from LinzerFH)
1008	U.S. Patent No. 6,646,578 (“Au”)
1009	Nokia Technologies Oy v. Amazon.com, Inc. et al., Case No. 1:23-cv-01236, ECF No. 6, (D. Del. Nov. 3, 2023) (Summons served 10/31/2023)
1010	Nokia Corporation et al. v. Amazon.com, Inc. et al., Case No. 1:23-cv-01232, ECF No. 8, (D. Del. Nov. 3, 2023) (Summons served 10/30/2023)
1011	February 2, 2024 Email from Joshua Newcomer with subject line “1380: Nokia’s proposed constructions.”
1012	H.26L over IP and H.324 Framework (“VCEG-N52”), available at https://www.itu.int/wftp3/av-arch/video-site/0109_San/VCEG-N52.doc (excerpted from ’818FH)
1013	Analysis of fractional pixel interpolation for MC prediction (“VCEG-N55”), available at https://www.itu.int/wftp3/av-arch/video-site/0109_San/VCEG-N55.doc .
1014	Declaration of June Munford (“Munford”)
1015	Tekalp, A. (1995). Digital Video Processing
1016	Certain Video Capable Electronic Devices, Including Computers, Streaming Devices, Televisions, and Components

Exhibit	Reference
	and Modules Thereof, Inv. No. 337-TA-1380 (USITC Nov. 30. 2023), Exhibit 6A to Complaint
1017	Certain Video Capable Electronic Devices, Including Computers, Streaming Devices, Televisions, and Components and Modules Thereof, Inv. No. 337-TA-1380 (USITC Nov. 30. 2023), Exhibit 6B to Complaint
1018	U.S. Patent Publication No. 2004/0161033 (“Notoya”)
1019	William Stallings, Computer Organization and Architecture, Designing for Performance, 8th Edition (2010)
1020	U.S. Patent No. 7,227,901 (“Joch”)
1021	U.S. Patent No. 6,539,059 (“Sriram”)
1022	Video coding for low bit rate communication (ITU-T H.263 Series H: Audiovisual and Multimedia Systems, Infrastructure of audiovisual services – Coding of moving video)
1023	John Watkinson, The MPEG Handbook, Second Edition (2004)
1024	Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Second Edition (1988)
1025	Declaration of Dr. Clifford Reader (“Reader”)

LISTING OF CHALLENGED CLAIMS

Reference	Claim Limitation
Claim 6	
6[pre]	A method for decoding sequences of pictures from a bitstream, wherein parameters are defined in a parameter set and each picture comprises information of one or more slices, the method comprising:
6[a]	recognizing, in a decoder, a sequence parameter set and forming at least one sequence parameter pertaining to a sequence using the parameter set;
6[b]	recognizing, in the decoder, a picture parameter set and forming at least one first picture parameter value pertaining to a picture using the parameter set;
6[c]	forming, in the decoder, at least one second picture parameter value using information of a slice header, the at least one second picture parameter value remaining unchanged at least in all slice headers of one picture; and
6[d]	using, in the decoder, the at least one second picture parameter value in decoding.
Claim 7	
7	The method according to claim 6, wherein the slice header is included with a reference to a picture parameter set.
Claim 8	
8	The method according to claim 6, wherein the picture parameter set is included with a reference to a sequence parameter set.
Claim 9	

Reference	Claim Limitation
9	The method according to claim 6, wherein the second picture parameter value is selected from a group of information of a picture order count and a frame number.
Claim 11	
11[pre]	A decoder for decoding sequences of pictures from a bitstream, wherein parameters are defined in a parameter set and each picture comprises information of one or more slices, the decoder comprising:
11[a]	a processor configured for:
11[b]	recognizing a sequence parameter set for a sequence of pictures and forming at least one sequence parameter pertaining to a sequence using the parameter set for a picture;
11[c]	recognizing a picture parameter set and forming at least one first picture parameter value pertaining to a picture using the parameter set;
11[d]	forming at least one second picture parameter value using information of a slice header, the at least one second picture parameter value remaining unchanged at least in all slice headers of one picture; and
11[e]	using the at least one second picture parameter value in decoding.
Claim 15	
15[pre]	A receiving device comprising a decoder for decoding sequences of pictures from a bitstream, wherein parameters are defined in a parameter set and each picture comprises information of one or more slices, the decoder comprising:
15[a]	a processor configured for:

Reference	Claim Limitation
15[b]	recognizing a sequence parameter set for a sequence of pictures and forming at least one sequence parameter pertaining to a sequence using the parameter set;
15[c]	recognizing a picture parameter set for a picture and forming at least one first picture parameter value pertaining to a picture using the parameter set;
15[d]	forming at least one second picture parameter value using information of a slice header, the at least one second picture parameter value remaining unchanged at least in all slice headers of one picture; and
15[e]	using the at least one second picture parameter value in decoding.
Claim 21	
21[pre]	A computer readable storage medium embedded with a computer program comprising programming code for execution on a processor, the programming code for decoding sequences of pictures from a bitstream, wherein parameters are defined in a parameter set and each picture comprises information of one or more slices, and the programming code for:
21[a]	recognizing a sequence parameter set for a sequence of pictures and forming at least one sequence parameter pertaining to a sequence using the parameter set;
21[b]	recognizing a picture parameter set for a picture and forming at least one first picture parameter value pertaining to a picture using the parameter set;
21[c]	forming at least one second picture parameter value using information of a slice header, the at least one second picture parameter value remaining unchanged at least in all slice headers of one picture; and

Reference	Claim Limitation
21[d]	using the at least one second picture parameter value in decoding.
Claim 23	
23[pre]	A decoder for decoding sequences of pictures from a bitstream, wherein parameters are defined in a parameter set and each picture comprises information of one or more slices, the decoder comprising:
23[a]	means for recognizing a sequence parameter set for a sequence of pictures and forming at least one sequence parameter pertaining to a sequence using the parameter set for a picture;
23[b]	means for recognizing a picture parameter set and forming at least one first picture parameter value pertaining to a picture using the parameter set;
23[c]	means for forming at least one second picture parameter value using information of a slice header, the at least one second picture parameter value remaining unchanged at least in all slice headers of one picture; and
23[d]	means for using the at least one second picture parameter value in decoding.

I. INTRODUCTION

Amazon.com, Inc. and Amazon.com Services LLC (“Amazon” or “Petitioner”) requests *inter partes* review (“IPR”) of claims 6-9, 11, 15, 21, and 23 (“Challenged Claims”) of U.S. Patent 7,724,818.

II. STANDING

The ’818 patent is available for IPR and Petitioner is not barred or estopped from requesting IPR on the grounds herein. Petitioner files this Petition within one year of service on Amazon of PO’s district court complaints (EX-1009, EX-1010).

III. IDENTIFICATION OF CHALLENGE

A. Prior Art¹

Petitioner requests review and cancellation of the Challenged Claims under:

Prior Art
JVT-E022d7; publicly available as of October 22, 2002 and incorporated by reference into Linzer; prior art at least under §102(a) and (as incorporated into Linzer) §102(e)(2) ² (EX-1007)
Linzer (U.S. Patent 7,660,356), filed October 22, 2002, published April 8, 2004, issued Feb. 9, 2010; prior art at least under §102(e)(2) (EX-1005)

¹ References herein are prior art based on the earliest priority date on the ’818 patent’s face. If PO claims an earlier invention date in this proceeding, Petitioner reserves the right to respond to such argument.

² All references to 35 U.S.C. §§102, 103 are to the pre-AIA statutes.

Au (U.S. Patent 6,646,578), filed November 22, 2002, issued on November 11, 2003; prior art at least under §102(e)(2) (EX-1008)
VCEG-N52; publicly available as of September 18, 2001; prior art under at least §102(b) (EX-1012)
Sriram (U.S. Patent 6,539,059), filed March 2, 2000, issued March 25, 2003; prior art at least under §102(a) and §102(e)(2) (EX-1021)

B. Grounds

Ground		Claims	Prior Art
1	103	6–9, 11, 15, 21, 23	JVT-E022d7
2	103	6–9, 11, 15, 21, 23	Linzer
3	103	6–9, 11, 15, 21, 23	Au
4	103	6–9, 11, 15, 21, 23	VCEG-N52, Sriram

IV. OVERVIEW

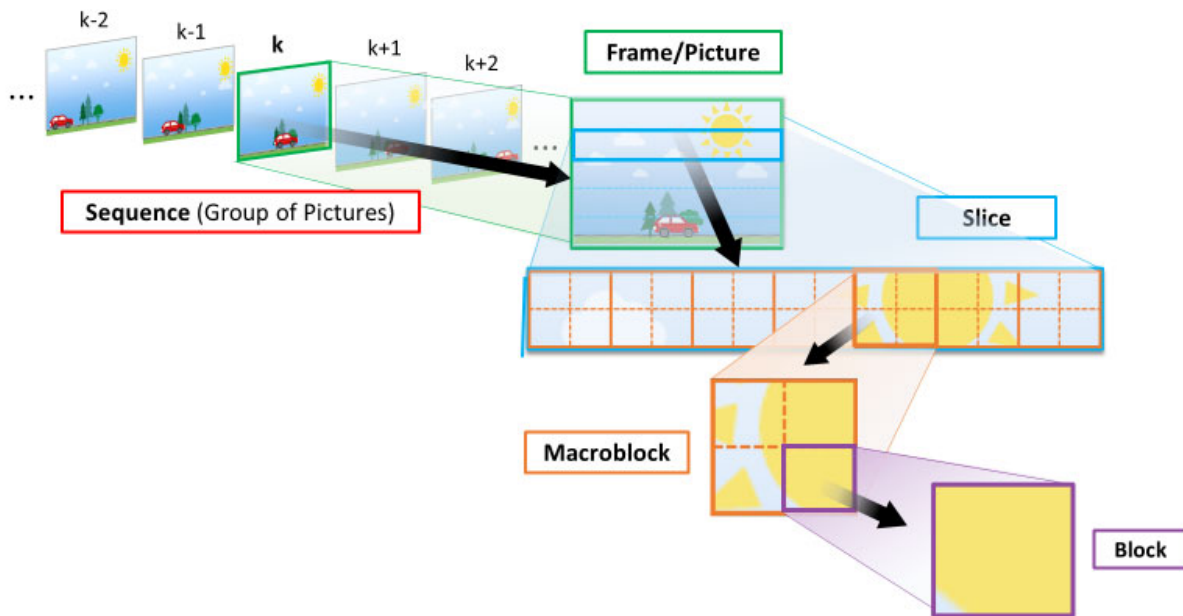
The '818 patent is titled “Method for coding sequences of pictures” and issued on May 25, 2010, from Application 10/426,928, filed on April 30, 2003.

A. Technology Background and Admitted Prior Art

Digital video involves sending and receiving video data in compressed form. EX-1015, 6. To process digital video, encoders and decoders have been widely used since at least 1995. Saber, ¶43 (citing EX-1015, 11-12).

The MPEG-1 standard (adopted in the 1990s) described digital video that followed a hierarchy, where a sequence comprises a group of pictures (GOP), a GOP

comprises pictures, each picture comprises slices, slices comprise macroblocks, and macroblocks comprise blocks (shown below). Saber, ¶44 (citing EX-1015, 10, 13).



The '818 patent's background describes the “parameter set concept,” which it acknowledges was described in VCEG-N52³, and describes known aspects of that concept. EX-1001, 2:48-3:28; EX-1012, 8; Saber, ¶45. According to the '818 patent, VCEG-N52 included a parameter set with “all picture, GOP, and sequence level data” (e.g., picture size, display window, optional coding modes employed,

³ Although the '818 patent refers to “VCEG-N55,” that is a typographical error and should refer to “VCEG-N52.” See EX-1001, References Cited; EX-1012; EX-1002, 146; see also EX-1013 (VCEG-N55 not related to parameters or parameter sets).

macroblock allocation map) and a unique identifier. EX-1001, 2:53-57. Further, in VCEG-N52, “[e]ach slice header includes a reference to a parameter set identifier, and the parameter values of the referred parameter set are used when decoding the slice.” *Id.*, 2:56-58. Importantly, the ’818 patent acknowledges VCEG-N52 discloses multiple parameter sets and transmitting a codeword indicating which parameter set to use in decoding. *Id.*, 3:5-10.

The ’818 patent discloses a conventional encoder and decoder that implement VCEG-N52’s known parameter set concept. *Id.*, 3:55-4:1; Saber, ¶46. Specifically, as Dr. Saber explains, Figure 2 depicts only conventional elements of an encoder/decoder system (encoder 1 and decoder 2); Figure 4 depicts only conventional decoder components (processor 2.2 and memory 2.3); and VCEG-N52’s parameter set concept was implemented on such a conventional encoder/decoder system. Saber, ¶46.

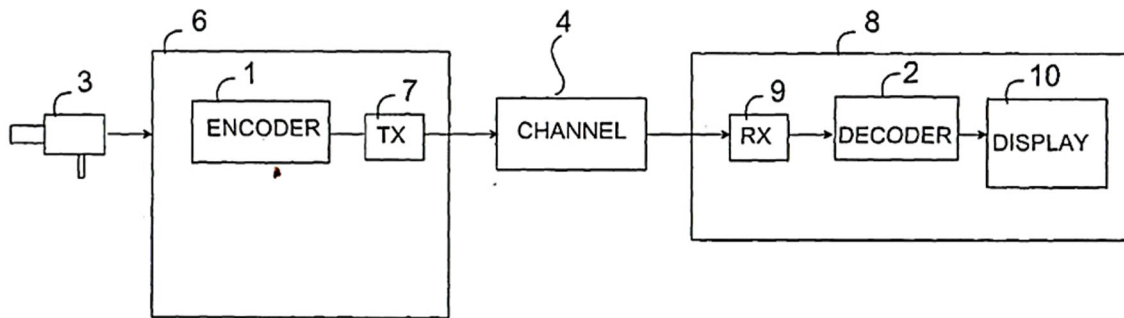


Fig. 2

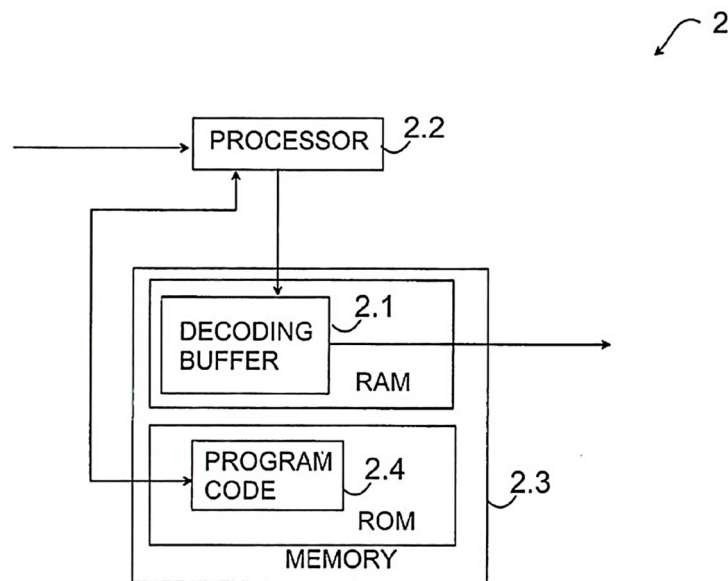


Fig. 4

B. Purported Invention

The '818 patent splits the admittedly-known parameter sets into “multiple parameter set structures according to the persistency and target of parameters.” EX-1001, 4:15-18. The parameter sets include “a sequence parameter set,” and “a picture parameter set,” as well as picture parameters in a “slice header.” *Id.*, Abstract, 4:41-48. The annotated image below shows how the parameter sets (independent GOP parameter set 105 (red), picture parameter set 104 (green), and slice parameter set 103 (blue)), illustrated in Figure 1 of the '818 patent, relate to a coded video sequence. Saber, ¶¶47-48.

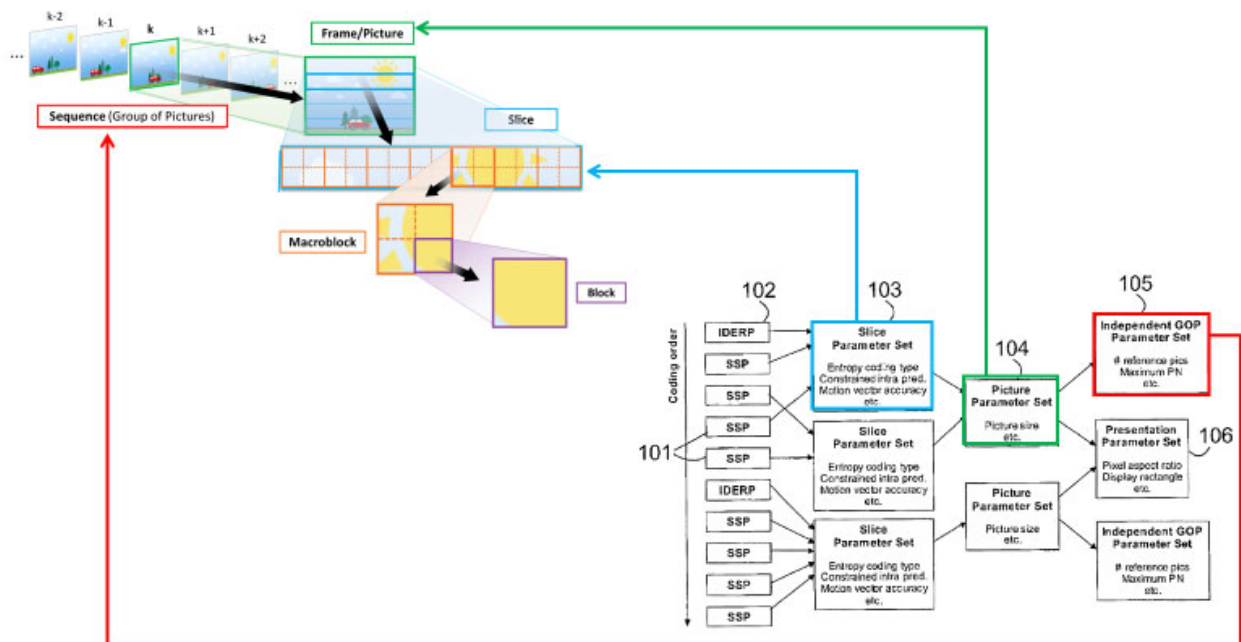


Fig. 1

C. Prosecution

During the '818 patent's prosecution, the Examiner repeatedly found the prior art taught the supposed point of novelty: the "separation of parameter sets to more than one category, including both a sequence parameter set and a picture parameter set." *E.g.*, EX-1002, 111, 142, 164, 166-67, 334-35, 363-64, 395-96; Saber, ¶50. Although the applicant repeatedly amended the claims to get around the prior art and ultimately gain allowance, this Petition demonstrates even those amended limitations were clearly in the prior art.

Originally-filed independent claim 6 recited "recognizing a sequence parameter set and forming at least one sequence parameter pertaining to a sequence using the parameter set; recognizing a picture parameter set and forming at least one first picture parameter pertaining to a picture using the parameter set; forming at least one second picture parameter using information of a slice header, the at least one second picture parameter remaining unchanged at least in all slice headers of one picture; and using the at least one second picture parameter in decoding." EX-1002, 31. Similar features were recited in other originally-filed independent claims. Saber, ¶51.

The Examiner issued five office actions, first rejecting the claims as anticipated by U.S. Patent Application 2004/0010802 ("Visharam") and later rejecting the independent claims as obvious over Visharam and Joch (EX-1020).

EX-1002, 111, 164, 334, 363, 395-96. The Examiner found Visharam disclosed a “sequence parameter set” and “picture parameter set,” even considering applicant’s argument that its invention was a “separation of parameter sets to more than one category, including both a sequence parameter set and a picture parameter set,” and relied on Joch for the picture parameter remaining unchanged in all slice headers of a picture. *Id.*, 142-43, 165, 178-87, 334, 351, -363-64, 381, 392, 395-96. During prosecution, the applicant argued Visharam disclosed “no more than what is included in” VCEG-N52, and provided VCEG-N52 to the Examiner. *Id.*, 142; Saber, ¶52.

Over six years after the application was filed, the applicant amended the pending independent claims corresponding to those challenged here to, *inter alia*, recite forming “picture parameter values” instead of forming “picture parameters,” including that the “second picture parameter value” (rather than second picture parameter) “remain[s] unchanged at least in all slice headers of one picture.” EX-1002, 403-04, 406-09. The applicant argued the references did not teach “defining at least one picture parameter value in a slice header, the picture parameter value remaining unchanged at least in all slice headers of one picture.” *Id.*, 404.

Following that, the Examiner allowed the claims without explanation. *Id.*, 417-20; Saber, ¶¶53-54.

V. LEVEL OF ORDINARY SKILL

A person of ordinary skill in the art (“POSITA”) at the time of the ’818 patent’s alleged invention 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 2-3 years of practical experience with video encoding/decoding. Additional experience can substitute for the level of education, and *vice-versa*. Saber, ¶¶22-23.

VI. CLAIM CONSTRUCTION

Claims should be construed “in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.” 37 C.F.R. §42.100(b). Petitioner is unaware of any “prior claim construction determination” related to the ’818 patent. Petitioner has not proposed any terms for construction because no constructions are necessary to resolve any disputes identified in this Petition.⁴ *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999).

⁴ Petitioner reserves its right to address construction of any terms raised at trial, or in the related actions.

A. Means-Plus-Function (MPF)⁵

When a claim limitation uses the term “means” and includes functional language, it is presumed to be a MPF term. *See* MPEP §2181. That presumption is overcome only if the limitation further includes the structure, material, or acts necessary to perform the recited function. *TriMed, Inc. v. Stryker Corp.*, 514 F.3d 1256, 1259-60 (Fed. Cir. 2008).

Claim 23 recites many limitations as “means for” performing a function. Petitioner identifies “a decoder as described in the ’818 patent specification in figure 4 and equivalents thereof” as the structure corresponding to the claimed functions of each “means for” limitation, and Petitioner notes this is the same structure PO alleged in the co-pending ITC investigation.⁶ *Saber*, ¶57; *see also* EX-1011.

⁵ Petitioner reserves the right to raise and address indefiniteness, enablement, and/or written description issues for the Challenged Claims in other proceedings.

⁶ Petitioner reserves the right to argue in other proceedings that the corresponding structure is insufficient, but for the reasons described in the Grounds in this Petition, the references teach this corresponding structure.

VII. DETAILED EXPLANATION OF GROUNDS

A. Ground 1: Obviousness over JVT-E022d7

1. JVT-E022d7

JVT-E022d7 qualifies as a printed publication, publicly available before the earliest priority date on the '818 patent's face. Munford, ¶¶4-17; Reader, ¶¶28-50. Both Ms. Munford and Dr. Reader provide supporting testimony and evidence, showing JVT-E022d7's printed publication status. Munford, ¶¶4 (reference "Generated: 2002-09-19," for an October 2002 meeting in Geneva, Switzerland), 5 (metadata indicating a September 19, 2002 last modified date), 9-16 (article referring to reference as published "Sept. 2002" and reference submitted to the patent office on October 22, 2002 by the Linzer applicant); Reader ¶42 ("Based on my personal experience and recollection, JVT-E022d7 was distributed and publicly available in September 2002."). Therefore, JVT-E022d7 was publicly available by October 22, 2002 and is prior art under §102(a). Munford, ¶17; Reader, ¶42.

JVT-E022d7 was an editor's proposed draft of what would become the H.264 standard. JVT-E022d7, 1. JVT-E022d7 discloses the same components of a video stream (bitstream includes sequences of pictures; a picture sequence corresponds to a group of pictures; a picture is a frame of a field; a picture consists of one or more slices) as discussed above and in the '818 patent (Section IV.A), and that all slices are part of a picture. *Id.*, 16, 20-21, 23, 27; Saber, ¶¶58-60.

JVT-E022d7 also describes the '818 patent's purported invention: splitting the parameters sets into the "sequence parameter set" and the "picture parameter set," as well as picture parameter values in a "slice header." Saber, ¶61. JVT-E022d7 discloses the use of "sequence parameter set[s]," which contain values that are "replaced only when an IDR [instantaneous decoder refresh] NAL [network abstraction layer] unit refers to a different sequence parameter set." JVT-E022d7, 52. In other words, like the '818 patent, each JVT-E022d7 sequence parameter set is used for a sequence of pictures. Saber, ¶61. And like the '818 patent, JVT-E022d7 discloses the use of "picture parameter set[s]," which "include[] the parameters that remain unchanged within a coded picture," and a "slice header" that is "part of a coded slice containing the coded representation of data elements pertaining to the slice data that follow the slice header." JVT-E022d7, 23, 52. Each slice header includes a parameter, "frame_num," which identifies the frame for which the slice is included and the value of which would not change for all slice headers in a picture. *Id.*, 57; Saber, ¶61.

2. Claim 6

a. 6[pre]

If limiting,⁷ JVT-E022d7 discloses or suggests 6[pre]. Saber, ¶¶62-65. JVT-E022d7 describes “an industry standard for compressed video representation” which includes a “decoding process” (i.e., a “method for decoding”). JVT-E022d7, 14, 16; Saber, ¶63.

As to “decoding sequences of pictures from a bitstream,” JVT-E022d7 defines “bitstream” as a “sequence of bits that forms the representation of data and coded fields and frames.” JVT-E022d7, 19. JVT-E022d7 discloses “requirements for the [network abstraction layer] NAL unit deliver and decoding order” and that “[d]ecoders conforming to this Recommendation | International Standard shall be capable of receiving NAL units in decoding order.” *Id.*, 70. JVT-E022d7 further defines “decoding order” as “[t]he order in which the coded pictures are to be decoded,” and states “[t]he decoding order is specified such that the decoding of pictures that use interpicture prediction follows later in decoding order than other pictures that are referenced in the decoding process.” *Id.*, 15, 20. In other words, JVT-E022d7 describes decoders that receive NAL units, which are independently

⁷ Petitioner reserves the right to argue the preamble of each Challenged Claim is limiting in other proceedings.

decodable components of a bitstream, in a decoding order, where the bitstream includes a sequence of bits forming fields and frames. Saber, ¶63

As to “parameters are defined in a parameter set,” in JVT-E022d7, “Sequence Parameter Set” and “Picture Parameter Set” are types of elements contained in the NAL units. JVT-E022d7, 50. Regarding the sequence parameter set, JVT-E022d7 discloses the “decoding order of a sequence parameter set shall precede the decoding order of other NAL units that refer to that sequence parameter set.” *Id.*, 70. Regarding the picture parameter set, JVT-E022d7 discloses a “picture parameter set includes the parameters that remain unchanged within a coded picture” and that the “decoding order of a picture parameter set shall precede the decoding order of other NAL units that refer to that picture parameter set.” *Id.*, 52, 70. Accordingly, a POSITA would have understood the sequence parameter set and picture parameter set have parameters defined in one or more parameter sets and are used in decoding of picture sequences because JVT-E022d7 describes the decoding order for these parameters sets. Saber, ¶64.

Further, in JVT-E022d7, “each picture comprises information of one or more slices”; “pictures are divided into macroblocks”; a slice contains an “integer number of macroblocks ordered contiguously in raster scan order within a particular slice group”; and each picture, therefore, includes one or more slices, where each slice includes macroblocks. JVT-E022d7, 23, 29. In JVT-E022d7, “[d]ecoding of a new

picture is started from the slice to be decoded.” *Id.*, 71. Based on this, a POSITA would have understood each picture comprises information of one or more slices, and decoding a picture includes decoding a slice of the picture. Saber, ¶65.

b. 6[a]

JVT-E022d7 discloses or suggests 6[a]. Saber, ¶¶66-70. JVT-E022d7 discloses use of a “sequence parameter set,” which contains values that are “replaced only when an IDR [instantaneous decoder refresh] NAL [network abstraction layer] unit refers to a different sequence parameter set.” JVT-E022d7, 52. JVT-E022d7 also provides examples of parameters for a sequence parameter set in raw byte sequence payloads (RBSP) syntax in Table 7.3.2.1. The sequence parameter set includes parameters providing information about properties and characteristics of a video sequence, such as picture order (“pic_order_cnt_type”), number of reference frames (“num_of_ref_frames”), picture size (“pic_width_in mbs_minus1” and “pic_height_in mbs_minus1”). *Id.*, 35. A POSITA would have understood JVT-E022d7 discloses using a sequence parameter set in decoding because the sequence parameter set includes values needed in decoding sequences, and such values are specific for a sequence parameter set. Saber, ¶66.

7.3.2 Raw byte sequence payloads and Rbsp trailing bits syntax

7.3.2.1 Sequence parameter set Rbsp syntax

Parameter	Category	Descriptor
seq_parameter_set_rbsp() {		
profile_idc	0	ue(v)
level_idc	0	ue(v)
seq_parameter_set_id	0	ue(v)
log2_max_frame_num_minus4	0	ue(v)
pic_order_cnt_type	0	ue(v)
if (pic_order_cnt_type == 0) {		
log2_max_pic_order_cnt_minus4	0	ue(v)
else if (pic_order_cnt_type == 1) {		
offset_for_non_stored_pic	0	se(v)
if (pic_order_cnt_type == 0) {		
frames_in_pic_order_cnt_cycle	0	ue(v)
if (frames_in_pic_order_cnt_cycle > 0) {		
for (i = 0; i < frames_in_pic_order_cnt_cycle; i++) {		
pred_frame[i]	0	se(v)
}		
}		
num_of_set_frames	0	ue(v)
required_frame_num_update_behaviour	0	u(1)
required_frame_num_update_behaviour_flag	0	u(1)
pic_width_in_mbs_minus1	0	ue(v)
pic_height_in_mbs_minus1	0	ue(v)
send_filter_parameters_flag	0	u(1)
constrained_intra_pred_flag	0	u(1)
mb_frame_field_adaptive_flag	0	u(1)
vui_seq_parameters_flag	0	u(1)
if (vui_seq_parameters_flag) {		
vui_seq_parameters()	0	
}		
rbsp_trailing_bits()	0	
}		

DRAFT ITU-T Rec. H.264 (2002 E) 17

JVT-E022d7, 35 (annotated).

The sequence parameter syntax in Table 7.3.2.1 includes a parameter, “Log2_max_frame_num_minus4,” (annotated red) which “specifies the MAX_FN [maximum frame number] used in frame number related arithmetic.” *Id.*, 52. “Log2_max_frame_num_minus4” is a parameter of the sequence parameter set that determines a range of frame numbers that can be used in a video sequence. Saber, ¶¶67-69 (citing JVT-E022d7, 35, 71-72, Table 7.3.2.1) (explaining the parameter “pic_order_cnt_type” is used for displaying pictures in a picture sequence in a correct order).

A POSITA would have understood that “recognizing, in a decoder, a sequence parameter set” (6[a]) encompasses a decoder parsing the incoming bitstream and

identifying a “sequence parameter set.” *Id.*, ¶70. JVT-E022d7 performs such “recognizing,” depending on the implementation, because its decoder extracts data elements (such as the sequence parameter set) from the bitstream based on the syntax described in the document or the sequence parameter set is *a priori* predefined in the decoder and the decoder recognizes a reference to the sequence parameter set from the bitstream. JVT-E022d7, 14-15; Saber, ¶70. JVT-E022d7 also discloses copying the decoded parameter sets into locations in the decoder. JVT-E022d7, 53. A POSITA would have further understood this parsing, extraction, and copying process results in “forming” (e.g., creating in local memory) the data element, such as, for this limitation “at least one sequence parameter pertaining to a sequence using the parameter set.” Saber, ¶70. A POSITA would have understood JVT-E022d7 teaches the “forming” limitation for the same reasons discussed above, including that it decodes pictures using, e.g., a maximum number of reference frames, a picture order count, and a flag indicating missing frame numbers. *Id.*

c. 6[b]

JVT-E022d7 discloses or suggests 6[b]. *Id.*, ¶¶71-74. First, JVT-E022d7 discloses the use of a “picture parameter set,” which “includes the parameters that remain unchanged within a coded picture.” JVT-E022d7, 52. JVT-E022d7 also discloses “[e]very picture parameter set shall refer to the active sequence parameter set [and] [t]he picture parameters of an active picture parameter set shall be replaced

only when the first slice NAL unit or DPA [Data Partition Type A] NAL unit of a picture refers to a different picture parameter set.” *Id.* JVT-E022d7 further discloses the “decoding order of a picture parameter set shall precede the decoding order of other NAL units that refer to that picture parameter set.” *Id.*, 70. A POSITA would have understood JVT-E022d7 discloses using a picture parameter set in decoding because JVT-E022d7 discloses decoding such parameter sets before decoding other NAL units that refer to that parameter set. Saber, ¶71. JVT-E022d7 provides examples of parameters for its picture parameter set in RBSP syntax in Table 7.3.2.2.

7.3.2.2 Picture parameter set RBSP syntax

	Category	Descriptor
pic_parameter_set_rbsp() {		
pic_parameter_set_id	1	ue(v)
seq_parameter_set_id	1	ue(v)
entropy_coding_mode	1	ue(v)
motion_resolution	1	ue(v)
adaptive_block_size_transform_flag	1	u(1)
num_slice_groups_minus1	1	ue(v)
if(num_slice_groups_minus1 > 0) {		
mb_allocation_map_type	1	ue(v)
if(mb_allocation_map_type == 0)		
for(i = 0; i <= num_slice_groups_minus1; i++)		
run_length	1	ue(v)
else if(mb_allocation_map_type == 2)		
for(i = 0; i < num_mbs_in_pic; i++)		
slice_group_id [Ed. Note: how many bits ?]	1	u(v)
else if(mb_allocation_map_type == 3)		
for(i = 0; i < num_slice_groups_minus1; i++) {		
top_left_mb	1	u(v)
bottom_right_mb	1	u(v)
}		
else if(mb_allocation_map_type == 4		
mb_allocation_map_type == 5		
mb_allocation_map_type == 6) {		
slice_group_change_direction	1	u(1)
slice_group_change_rate_minus1	1	ue(v)
}		
}		
num_ref_idx_l0_active_minus1	1	ue(v)
num_ref_idx_l1_active_minus1	1	ue(v)
weighted_pred_flag	1	u(1)
weighted_bipred_explicit_flag	1	u(1)
weighted_bipred_implicit_flag	1	u(1)
slice_qp_minus26 /* relative to 26 */	1	se(v)
slice_qp_s_minus26 /* relative to 26 */	1	se(v)
redundant_slice_flag	1	u(1)
vui_pic_parameters_flag	1	u(1)
if(vui_pic_parameters_flag) {		
vui_pic_parameters()	1	
}		
rbsp_trailing_bits()	1	
}		

JVT-E022d7, 36 (annotated). The syntax in Table 7.3.2.2 includes as “picture parameters,” “num_slice_groups_minus1,” “num_ref_idx_l0_active_minus1,” and “weighted_pred_flag” (all annotated green). *Id.* Each of these teaches a “first picture parameter value pertaining to a picture using the parameter set.” Saber, ¶72 (citing JVT-E022d7, 54 (conditions where “num_slice_groups_minus1” is zero or greater than zero), 55 (“Num_ref_idx_l0_active_minus1” “specifies the number of

reference pictures minus 1 in the reference list 0 that are used to decode the picture” and “weighted_pred_flag” with a value of 0 or 1)).

A POSITA would have understood “recognizing, in the decoder” (6[b]) encompasses a decoder parsing the incoming bitstream and identifying a “picture parameter set.” Saber, ¶73. JVT-E022d7 performs such “recognizing” because, depending on the implementation, its decoder extracts data elements (such as the picture parameter set) from the bitstream based on the described syntax or the picture parameter set that is *a priori* predefined in the decoder and the decoder recognizes a reference to the picture parameter set from the bitstream. JVT-E022d7, 14-15; Saber, ¶73. A POSITA would have further understood this parsing and extraction process results in “forming” (e.g., creating in local memory) the value associated with the data element, such as, for this limitation “at least one first picture parameter value pertaining to a picture using the parameter set.” Saber, ¶¶73-74.

d. 6[c]

JVT-E022d7 discloses or suggests 6[c]. Saber, ¶¶75-78. Initially, JVT-E022d7 teaches “pictures” as well as “slice headers.” JVT-E022d7, 20-21, 23; Saber, ¶75. JVT-E022d7 also discloses or suggests “forming, in the decoder, at least one second picture parameter value using information of a slice header.” Saber, ¶76. JVT-E022d7 provides examples of parameters for a slice header syntax in Table 7.3.3, including “frame_num” (annotated blue).

7.3.3 Slice header syntax

frame_num			
slice_header() {		Category	Descriptor
pic_parameter_set_id	4		ue(v)
frame_num	4		u(v)
pic_structure	4		ue(v)
first_mb_in_slice	4		u(v)
slice_type_idc	4		ue(v)
if(pic_order_cnt_type == 0)			
pic_order_cnt	4		u(v)
else if(pic_order_cnt_type == 1)			
delta_pic_order_cnt	4		se(v)
if(redundant_slice_flag)			
redundant_pic_cnt	4		ue(v)
if(slice_type_idc == BiPred)			
direct_spatial_mv_pred_flag	4		u(1)
num_ref_idx_active_override_flag	4		u(1)
if(num_ref_idx_active_override_flag) {			
if(slice_type_idc == Pred slice_type_idc == SPred slice_type_idc == BiPred) {			
num_ref_idx_l0_active_minus1	4		ue(v)
if(slice_type_idc == BiPred)			
num_ref_idx_l1_active_minus1	4		ue(v)
}			
}			

JVT-E022d7, 40 (annotated). “frame_num serves as a unique ID for each frame stored in the reference picture buffer,” and “frame_num shall be incremented by 1 for each coded picture in decoding order.” *Id.*, 57. Thus, “frame_num” refers to a value that changes only from picture to picture. Saber, ¶77. A POSITA would have understood the “frame_num” value serves as a unique identifier for each frame and remains the same for all the slice headers of a frame (i.e., a frame corresponding to a picture). *Id.*; JVT-E022d7, 57. If it changed, the decoder would end up decoding the slice into the wrong frame, resulting in an image that did not match the original. Saber, ¶77. Therefore, JVT-E022d7’s disclosure of a slice header identifying the current frame teaches “one second picture parameter value remaining unchanged at least in all slice headers of one picture” (6[c]). *Id.*

A POSITA would have understood “forming, in the decoder, at least one second picture parameter value using information of a slice header” (6[c]) encompasses parsing the incoming bitstream and identifying a second picture parameter value in the recited “slice header” (“using information of a slice header”) and saving that value in local memory (“forming”). Saber, ¶78. A POSITA would have understood JVT-E022d7 teaches this “forming” limitation for the same reasons discussed above, including that it describes extracting data elements (such as the slice header) from the bitstream based on the syntax described in the document and because the value of `frame_num` is saved in local memory along with the reference picture. *Id.*; JVT-E022d7, 14-15.

e. 6[d]

JVT-E022d7 discloses or suggests 6[d]. Saber, ¶79. JVT-E022d7 discloses the decoder uses the “`frame_num`” value (i.e., “the at least one second picture parameter value”) in decoding because “`frame_num` shall be incremented by 1 for each coded picture in decoding order...[t]he decoding order of primary coded pictures shall be non-decreasing in frame number order...`frame_num` serves as a unique ID for each frame stored in the reference picture buffer [and a] decoder...encounters a frame number.” JVT-E022d7, 57. A POSITA would have understood `frame_num`, which is a unique ID for a given frame, when present is used in decoding a sequence of frames to identify each frame. Saber, ¶79. Hence, this

unique ID is used to properly index and decode each frame in a given sequence when present. *Id.*

3. Claim 7

Section VII.A.2 explains JVT-E022d7 discloses or suggests the “method according to claim 6.” As discussed in Section 2.d (6[c]), JVT-E022d7 discloses a slice header is “[a] part of a coded slice containing the coded representation of data elements pertaining to the slice data that follow the slice header.” JVT-E022d7, 23. JVT-E022d7 discloses “pic_parameter_set_id” (annotated blue below) in the slice header syntax and a data element in the picture parameter set syntax . *Id.*, 36 (annotated below), 40; Saber, ¶80. Therefore, JVT-E022d7’s slice header includes “pic_parameter_set_id,” which is a reference to a picture parameter set. Saber, ¶80.

7.3.3 Slice header syntax

	Category	Descriptor
slice_header() {		
pic_parameter_set_id	4	ue(v)
frame_num	4	u(v)
pic_structure	4	ue(v)
first_sub_in_slice	4	u(v)
slice_type_idc	4	ue(v)
if(pic_order_cnt_type == 0)		
pic_order_cnt	4	u(v)
else if(pic_order_cnt_type == 1)		
delta_pic_order_cnt	4	se(v)
if(redundant_slice_flag)		
redundant_pic_cnt	4	ue(v)
if(slice_type_idc == BiPred)		
direct_spatial_mv_pred_flag	4	u(1)
num_ref_idx_active_override_flag	4	u(1)
if(num_ref_idx_active_override_flag) {		
if(slice_type_idc == Pred slice_type_idc == SPred)		
slice_type_idc == BiPred) {		
num_ref_idx_l0_active_minus1	4	ue(v)
if(slice_type_idc == BiPred)		
num_ref_idx_l1_active_minus1	4	ue(v)
}		
}		

pic_parameter_set_id

4. Claim 8

Section VII.A.2 explains JVT-E022d7 discloses or suggests the “method according to claim 6.” As discussed in Section 2.c (6[b]), JVT-E022d7’s Table 7.3.2.2 discloses a data element in a picture parameter syntax is “seq_parameter_set_id” (annotated green below), which “identifies the sequence parameter set to be referred” in a sequence. JVT-E022d7, 36 (annotated below), 52. Therefore, JVT-E022d7’s picture parameter set includes “seq_parameter_set_id,” which is a reference to a sequence parameter set. Saber, ¶81.

7.3.2.2 Picture parameter set RBSP syntax

	Category	Descriptor
pic_parameter_set_rbsp() {		
pic_parameter_set_id	1	ue(v)
seq_parameter_set_id	1	ue(v)
entropy_coding_mode	1	ue(v)
motion_resolution	1	ue(v)
adaptive_block_size_transform_flag	1	u(1)
num_slice_groups_minus1	1	ue(v)
if(num_slice_groups_minus1 > 0) {		
mb_allocation_map_type	1	ue(v)
if(mb_allocation_map_type == 0)		
for(i = 0; i <= num_slice_groups_minus1; i++)		
run_length	1	ue(v)
else if(mb_allocation_map_type == 2)		
for(i = 0; i < num_mbs_in_pic; i++)		
slice_group_id [Ed. Note: how many bits ?]	1	u(v)
else if(mb_allocation_map_type == 3)		
for(i = 0; i < num_slice_groups_minus1; i++) {		
top_left_mb	1	u(v)
bottom_right_mb	1	u(v)
}		
else if(mb_allocation_map_type == 4		
mb_allocation_map_type == 5		
mb_allocation_map_type == 6) {		
slice_group_change_direction	1	u(1)
slice_group_change_rate_minus1	1	ue(v)
}		
}		
num_ref_idx_l0_active_minus1	1	ue(v)
num_ref_idx_l1_active_minus1	1	ue(v)
weighted_pred_flag	1	u(1)
weighted_bipred_explicit_flag	1	u(1)
weighted_bipred_implicit_flag	1	u(1)
slice_qp_minus26 /* relative to 26 */	1	se(v)
slice_qp_s_minus26 /* relative to 26 */	1	se(v)
redundant_slice_flag	1	u(1)
vui_pic_parameters_flag	1	u(1)
if(vui_pic_parameters_flag) {		
vui_pic_parameters()	1	
}		
rbsp_trailing_bits()	1	
}		

5. Claim 9

Section VII.A.2 explains JVT-E022d7 discloses or suggests the “method according to claim 6.” As discussed in Section VII.A.2.d (6[c]), the value of JVT-E022d7’s “frame_num” parameter teaches a “second picture parameter value” and “frame_num” identifies the frame number and so is “selected from a group of information of a picture order count and a frame number.” JVT-E022d7, 40, 57; Saber, ¶82.

6. Claim 11

a. 11[pre]

If limiting, JVT-E022d7 discloses or suggests the preamble. Saber, ¶83. As discussed in VII.A.2.a (6[pre]), JVT-E022d7 discloses or suggests “decoding sequences of pictures from a bitstream, wherein parameters are defined in a parameter set and each picture comprises information of one or more slices.” JVT-E022d7 further discloses that such decoding is done by a decoder. JVT-E022d7, 17, 145, Fig. 11-1; Saber, ¶83.

b. 11[a]

JVT-E022d7 discloses or suggests 11[a]. A POSITA would have understood that JVT-E022d7 discloses a processor for decoding a video, including the steps in 11[b]-11[e], because a processor performs the manipulation (including decoding) of

motion video in the form of computer data so that the manipulated computer data can be stored on various storage media. Saber, ¶84 (citing JVT-E022d7, 14).

c. 11[b]

As discussed in Section VII.A.2.b (6[a]), JVT-E022d7 discloses or suggests “recognizing, in a decoder, a sequence parameter set and forming at least one sequence parameter pertaining to a sequence using the parameter set.” 11[b] is substantially similar to 6[a] except 11[b] recites “recognizing a sequence parameter set for a sequence of pictures [and] using the parameter set for a picture.” JVT-E022d7 teaches these 11[b] additions because JVT-E022d7 discloses a “video source that is represented by the bitstream is a sequence of frames and/or fields (called collectively pictures) in decoding order.” JVT-E022d7, 27. A POSITA would have understood the sequence parameter is “for a sequence of pictures” (11[b]) because a video in a bitstream is a sequence of pictures in decoding order and the sequence parameters apply to one or more pictures in a coded video sequence. Saber, ¶85. A POSITA would have also understood at least some of the parameters in JVT-E022d7’s sequence parameter set, including “pic_order_cnt_type” as explained in 6[a], would be “for a picture” because that parameter is used for displaying pictures in a picture sequence in correct order and is therefore defined for each picture. *Id.*

d. 11[c]

JVT-E022d7 discloses or suggests 11[c] for substantially the same reasons as 6[b]. Section VII.A.2.c; Saber, ¶86.

e. 11[d]

JVT-E022d7 discloses or suggests 11[d] for substantially the same reasons as 6[c]. Section VII.A.2.d; Saber, ¶87.

f. 11[e]

JVT-E022d7 discloses or suggests 11[e] for substantially the same reasons as 6[d]. Section VII.A.2.e; Saber, ¶88.

7. Claim 15

a. 15[pre]

If limiting, JVT-E022d7 discloses or suggests 15[pre]. Saber, ¶¶89-90. 15[pre] adds only a “receiving device” comprising a decoder as in 11[pre]. Section VII.A.6.a. A POSITA would have understood, and found it obvious, that JVT-E022d7 uses a receiving device comprising a decoder. Saber, ¶89. (citing EX-1015, 8, 12; Au, 6:66-7:3, Fig. 1).

JVT-E022d7 “allows motion video to be manipulated as a form of computer data and to be stored on various storage media, transmitted and received over existing and future networks.” JVT-E022d7, 14; Saber, ¶90. A POSITA would have understood this discloses, or at least suggests, a “receiving device” because it

receives and stores data, and a POSITA would have also understood decoding in JVT-E022d7 is performed in such a device because the received data is what is decoded. Saber, ¶90.

b. 15[a]

JVT-E022d7 discloses or suggests 15[a] for substantially the same reasons as 11[a]. Section VII.A.6.b; Saber, ¶91.

c. 15[b]

JVT-E022d7 discloses or suggests 15[b] for substantially the same reasons as 11[b]. Section VII.A.6.c; Saber, ¶92.

d. 15[c]

JVT-E022d7 discloses or suggests 15[c] for substantially the same reasons as 11[c]. Section VII.A.6.d; Saber, ¶93.

e. 15[d]

JVT-E022d7 discloses or suggests 15[d] for substantially the same reasons as 11[d]. Section VII.A.6.e; Saber, ¶94.

f. 15[e]

JVT-E022d7 discloses or suggests 15[e] substantially the same reasons as 11[e]. Section VII.A.6.f; Saber, ¶95.

8. Claim 21

a. 21[pre]

If limiting, JVT-E022d7 discloses or suggests 21[pre]. Saber, ¶¶96-97. Beyond 6[pre] (Section VII.A.2.a) and 11[pre] (Section VII.A.6.a), 21[pre] adds only a “computer readable storage medium embedded with a computer program comprising programming code for execution on a processor” and that program is for the “decoding” similarly recited in 6[pre]. JVT-E022d7 discloses this addition.

As discussed in Section VII.A.6.a (11[pre]), JVT-E022d7’s decoding is done by a decoder. JVT-E022d7, 17, 145, Fig. 11-1; Saber, ¶97. And as discussed in Section VII.A.6.b (11[a]), a POSITA would have understood JVT-E022d7 discloses a processor for decoding a video. JVT-E022d7, 14; Saber, ¶97. A POSITA would have further understood such a decoder using a processor would have been implemented with programming code in a computer program stored on a computer readable storage medium. Saber, ¶97.

b. 21[a]

JVT-E022d7 discloses or suggests 21[a] for substantially the same reasons as 11[b]. Section VII.A.6.c; Saber, ¶98.

c. 21[b]

JVT-E022d7 discloses or suggests 21[b] for substantially the same reasons as 6[b]. Section VII.A.2.c; Saber, ¶99.

d. 21[c]

JVT-E022d7 discloses or suggests 21[c] for substantially the same reasons as 6[c]. Section VII.A.2.d; Saber, ¶100.

e. 21[d]

JVT-E022d7 discloses or suggests 21[d] for substantially the same reasons as 6[d]. Section VII.A.2.e; Saber, ¶101.

9. Claim 23

a. 23[pre]

If limiting, JVT-E022d7 discloses or suggests 23[pre] for substantially the same reasons as 11[pre]. Section VII.A.6.a; Saber, ¶102.

b. 23[a]

JVT-E022d7 discloses or suggests 23[a] regardless of whether it is construed as MPF. Saber, ¶¶103-06.

JVT-E022d7 teaches the function as discussed in Section VII.A.6.c (11[b]). *Id.*, ¶104.

If this term is construed as MPF, JVT-E022d7 discloses or suggests the structure discussed in Section VI.A (“a decoder as described in the ’818 patent specification in figure 4 and equivalents thereof”). *Id.*, ¶105. As shown below, the ’818 patent’s Figure 4 discloses a structure including a processor, decoding buffer, program code (decoding buffer and program code both stored in memory).

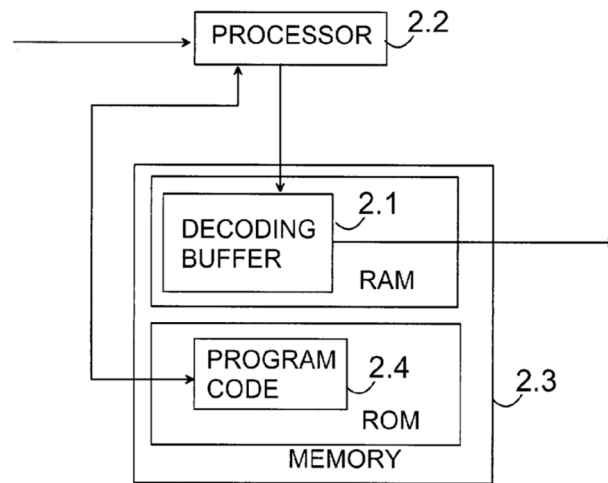


Fig. 4

As discussed in Section VII.A.6.c (11[b]), JVT-E022d7 discloses that its decoding is performed by a decoder. JVT-E022d7, 17, 145, Fig. 11-1. Section VII.A.6.b (11[a]) explains JVT-E022d7 discloses or suggests a processor configured to perform the recited steps that follow. Section VII.A.8.a (21[pre]) explains a POSITA would have understood, or found it obvious, that JVT-E022d7's decoder uses a processor implemented with programming code in a computer program stored on a computer readable storage medium. JVT-E022d7's decoder also includes a number of buffers, including "short and long term buffers" and a "reference picture buffer," which are managed using "Adaptive Memory Control." JVT-E022d7, 80.

A POSITA would have understood JVT-E022d7 discloses a decoder with structure the same as, or equivalent to, that of '818 patent Figure 4, and that decoder performs 23[a]'s recited function. Saber, ¶106.

c. 23[b]

JVT-E022d7 discloses or suggests 23[b] regardless of whether it is construed as MPF. *Id.*, ¶¶107-09.

JVT-E022d7 teaches the function as discussed in Section VII.A.2.c (6[b]). *Id.*, ¶108.

If this term is construed as MPF, JVT-E022d7's decoder discloses or suggests the structure discussed in Section VI.A ("a decoder as described in the '818 patent specification in figure 4 and equivalents thereof") for the same reasons discussed in 23[a] above and that decoder performs 23[b]'s recited function. *Id.*, ¶109.

d. 23[c]

JVT-E022d7 discloses or suggests 23[c] regardless of whether it is construed as MPF. *Id.*, ¶¶110-12.

JVT-E022d7 teaches the function as discussed in Section VII.A.2.d (6[c]). *Id.*, ¶111.

If this term is construed as MPF, JVT-E022d7's decoder discloses or suggests the structure discussed in Section VI.A ("a decoder as described in the '818 patent

specification in figure 4 and equivalents thereof”) for the same reasons discussed in 23[a] above and that decoder performs 23[c]’s recited function. *Id.*, ¶112.

e. 23[d]

JVT-E022d7 discloses or suggests 23[d] regardless of whether it is construed as MPF. *Id.*, ¶¶113-15.

JVT-E022d7 teaches the function as discussed in Section VII.A.2.e (6[d]). *Id.*, ¶114.

If this term is construed as MPF, JVT-E022d7’s decoder discloses or suggests the structure discussed in Section VI.A (“a decoder as described in the ’818 patent specification in figure 4 and equivalents thereof”) for the same reasons discussed in 23[a] above and that decoder performs 23[d]’s recited function. *Id.*, ¶115.

B. Ground 2: Obviousness over Linzer

1. Linzer

Linzer relates to a “method for decoding a digital video bit-stream.” Linzer, Abstract. In its “Background of the Invention,” Linzer “incorporated by reference” the “entirety” of JVT-E022d7 (JVT-E022d7, included in EX-1006, 47-264). Linzer, 1:27-35; Saber, ¶116. As such, Linzer discloses JVT-E022d7 as if it were fully contained within Linzer. The Linzer applicant filed JVT-E022d7 in its entirety with the Linzer application. EX-1006, 47. “[I]nformation incorporated is as much a part of the application as filed as if the text was repeated in the application, and should

be treated as part of the text of the application as filed.” MPEP §2163.07(b). Material incorporated by reference can be used for purposes of unpatentability when a host document “identifies the material subject to incorporation (the entirety of the identified applications, not just particular portions thereof).” *Neumodx Molecular, Inc. v. Handylab, Inc.*, IPR2020-01133, Paper 23, 5-6 (Aug. 6, 2021); *Advanced Display Sys., Inc. v. Kent State Univ.*, 212 F.3d 1272, 1282 (Fed. Cir. 2000) (“Material not explicitly contained in the single, prior art document may still be considered for purposes of anticipation if that material is incorporated by reference into the document.”).

2. Claims 6-9, 11, 15, 21, and 23

If PO disputes the printed publication status of JVT-E022d7 for Ground 1, JVT-E022d7’s disclosure is also prior art under §102(e)(2) because it was incorporated by reference in its entirety in Linzer. Claims 6-9, 11, 15, 21, and 23 are unpatentable as obvious over Linzer based on the same disclosures of JVT-E022d7 (incorporated by reference into Linzer) as discussed in Section VII.A. Saber, ¶117.

C. Ground 3: Obviousness over Au

1. Au

Au relates to “a system and method for context adaptive entropy decoding of transform coefficients in compressed video images.” Au, 1:7-9; Saber, ¶¶118-21.

Part of Au's system and method involves "a hierarchy of syntax levels, with the 3 main levels being a sequence level **17**, a picture (or frame) level **19**, and slice level **21**." Au, 8:7-9. Au also discloses "parameter sets," that "allow[] efficient transmission of infrequently changing data at the sequence **17** and picture level **19** in the H.264 standard." *Id.*, 8:9-11. Each sequence parameter set "includes values of parameters that will remain unchanged for an entire video sequence, or from one instantaneous decoder refresh (IDR) picture to the next." *Id.*, 8:13-15. Each picture parameter set "includes values of parameters that will remain unchanged within a coded representation of a picture (frame or field) **22**." *Id.*, 8:20-22. In addition, Au discloses a "slice header," which contains "a coded representation of data elements **35** that pertain to the decoding of the slice data that follow the slice header **27**." *Id.*, 8:40-42. The hierarchy, parameter sets, and slice header are illustrated in Figure 2 of Au, reproduced below.

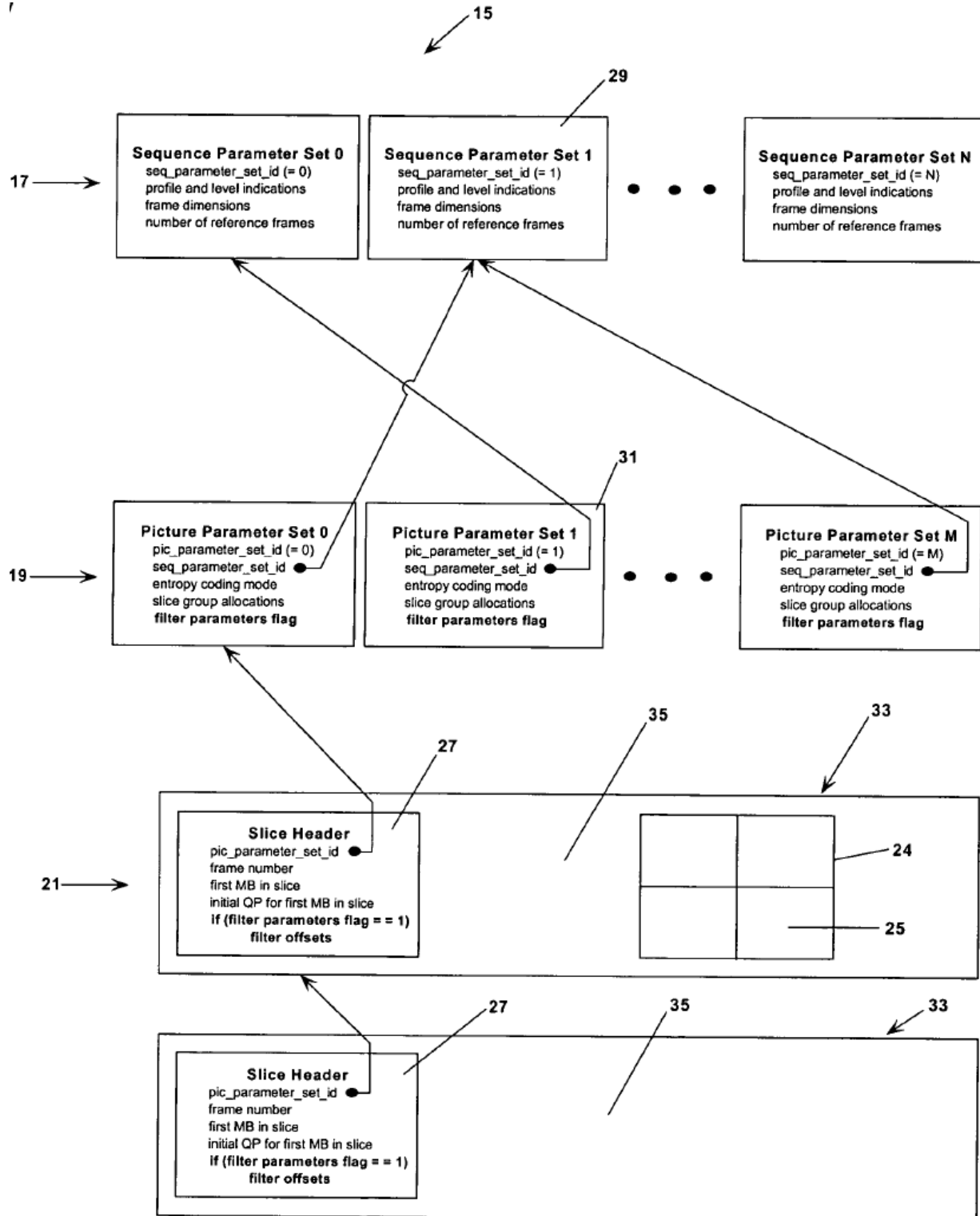


Figure 2

As shown, each slice and picture parameter set indicates the parameter set next up in the hierarchy by including the identifier for that parameter set. *Id.*, 9:12-14.

Au's hierarchy of syntax levels corresponds to the background disclosure in the '818 patent, as explained in Section IV.A. Saber, ¶¶119-21. Au also describes the '818 patent's purported invention: splitting the parameter sets into the "sequence parameter set" and the "picture parameter set," as shown in Au's Figure 2 hierarchy above, as well as picture parameters (e.g., Au's "frame_num") included in a "slice header." *Id.*

2. Claim 6

a. 6[pre]

If limiting, Au discloses or suggests 6[pre]. Saber, ¶¶122-25. As to "decoding sequences of pictures from a bitstream," Au discloses a "method...for block-based motion-compensated decoding." Au, Abstract. Au further discloses a bitstream processed by a decoder to "produce reconstructed video images" which correspond to "reconstructed frames." *Id.*, 10:50-53. A POSITA would have understood such reconstruction to be decoding a sequence of pictures from the bitstream because frames are synonymous with pictures in modern digital video reconstruction. Saber, ¶122.

Au further discloses “parameter sets” that include “values of parameters.” Au, 8:9-19. A POSITA would have understood this to disclose “parameters are defined in a parameter set.” Saber, ¶123.

As to “each picture comprises information of one or more slices,” Au discloses a slice “contains picture data 35 representing a sub-set of the macroblocks 24 of the complete picture 22.” Au, 8:36-38. Therefore, each picture comprises information (“data...representing a sub-set of the macroblocks”) of one or more slices. Saber, ¶¶124-25.

b. 6[a]

Au discloses or suggests 6[a]. Saber, ¶¶126-28. Au discloses since the video data is transmitted or stored in the form of compressed bitstream, a decoder is needed to decode the bitstream to reconstruct the video. Au, 1:55-57. Au therefore describes a decoder. Saber, ¶126. Au discloses a “sequence parameter set” including “values of parameters that will remain unchanged for an entire video sequence, or from one instantaneous decoder refresh (IDR) picture to the next.” Au, 8:11-15. Au discloses examples of parameters in a sequence parameter set, including “frame dimensions and the maximum number of reference frames,” as well as “profile and level indications[.]” *Id.*, 8:16-19, Fig. 2. Au discloses the “active sequence parameters...are used when decoding coded pictures.” *Id.*, 8:30-

34, 9:12-14 (“The parameter sets **31** also contain a reference to the sequence parameter sets **29**, which are active for decoding of the pictures **22**.”).

A POSITA would have understood “recognizing, in a decoder” (6[a]) encompasses a decoder parsing the incoming bitstream and identifying the recited “sequence parameter set.” A POSITA would have understood “forming” (6[a]) encompasses the decoder creating in local memory the data element, such as, for this limitation “at least one sequence parameter pertaining to a sequence using the parameter set.” Saber, ¶127. A POSITA would have understood Au’s decoder parses the incoming bitstream to extract (“recognizing”) and store (“forming”) data elements, including the recited “sequence parameter set” or the sequence parameter set is *a priori* predefined in the decoder and the decoder recognizes a reference to the sequence parameter set from the bitstream described above. *Id.*, ¶¶127-28.

c. 6[b]

Au discloses or suggests 6[b]. Saber, ¶¶129-31. First, Au discloses a “picture parameter set,” including “values of parameters that will remain unchanged within a coded representation of a picture (frame or field) **22**.” Au, 8:20-22. Au discloses examples of parameters in a picture parameter set, including “the entropy coding mode and a flag that specifies whether deblocking filter parameters will be transmitted in the slice headers **27** of the picture **22**[,]” as well as “slice group allocations[.]” *Id.*, 8:23-26, Fig. 2. Au also discloses “each picture parameter set **31**

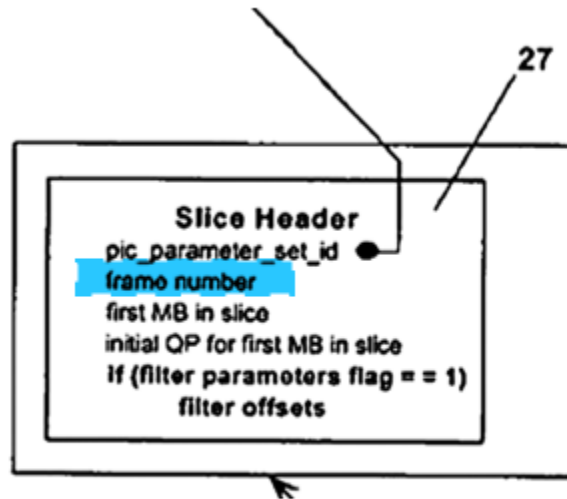
contains parameter values that pertain to the decoding of the pictures **22** for which the particular parameter set **31** is active (i.e., selected in the slice headers **27** of the picture **22**).” *Id.*, 9:8-12. Au decodes pictures (“in a decoder”) using, for example, entropy coding mode, a flag that specifies whether deblocking filter parameters will be transmitted in the slice headers, and slice group allocations (“forming at least one first picture parameter value pertaining to a picture using the parameter set”). Saber, ¶129.

A POSITA would have understood “recognizing, in the decoder” (6[b]) encompasses a decoder parsing the incoming bitstream and identifying the recited “picture parameter set,” and “forming” (6[b]) encompasses the decoder creating in local memory the value associated with the data element, such as, for this limitation “at least one first picture parameter value pertaining to a picture using the parameter set.” *Id.*, ¶130. A POSITA would have understood Au’s decoder parses the incoming bitstream to extract (“recognizing”) and store (“forming”) data elements, including the recited “picture parameter set” or the picture parameter set is *a priori* predefined in the decoder and the decoder recognizes a reference to the picture parameter set from the bitstream described above. *Id.*, ¶¶130-31.

d. 6[c]

Au discloses or suggests 6[c]. *Id.*, ¶¶132-35. Initially, Au teaches “pictures” (Au, 8:22) as well as “slice headers” (*id.*, 8:40-42). Specifically, Au discloses that

a “picture” is a “frame or field.” *Id.*, 8:22. Au’s slice header in Figure 2 (relevant portion reproduced below) includes “frame number” (annotated blue).



A POSITA would have understood Au’s frame number is a picture parameter value, and therefore teaches “at least one second picture parameter value” formed by the decoder using information from a slice header. Saber, ¶¶132-33.

Au also discloses or suggests “forming, in the decoder, at least one second picture parameter value using information of a slice header.” A POSITA would have understood “forming, in the decoder” (6[c]) encompasses a decoder creating in local memory the value associated with the data element, such as, for this limitation “at least one second picture parameter value using information of a slice header.” *Id.*, ¶134. A POSITA would have understood Au’s decoder parses the incoming bitstream to extract and store (“forming”) values associated with data elements in

order for them to be utilized in the decoding process, including the recited “second picture parameter value” described above. *Id.*

Au also discloses or suggests “the at least one second picture parameter value remaining unchanged at least in all slice headers of one picture.” Au discloses a slice “contains picture data **35** representing a sub-set of the macroblocks **24** of the complete picture **22**.” Au, 8:36-38. Au discloses a picture is a “frame or field.” *Id.*, 8:22; *see also id.*, 7:25-26. Therefore, Au discloses a picture that is a frame, the data of which is included in slices. Saber, ¶135. In such a case in which frame_num is used, each slice of the picture would have the same frame number to properly reconstruct the frame. *Id.* That frame number, identified above as the “second picture parameter value,” would thus “remain[] unchanged at least in all slices of one picture,” as required 6[c]. *Id.* frame_num functions this way to ensure all slices belonging to a given frame are indexed accordingly to spatially position them correctly within the frame for proper reconstruction. *Id.* Hence, Au’s slice header includes frame number (“at least one second picture parameter value using information of a slice header”), which is the same for all slices in the frame (“remaining unchanged at least in all slice headers of one picture”). *Id.*

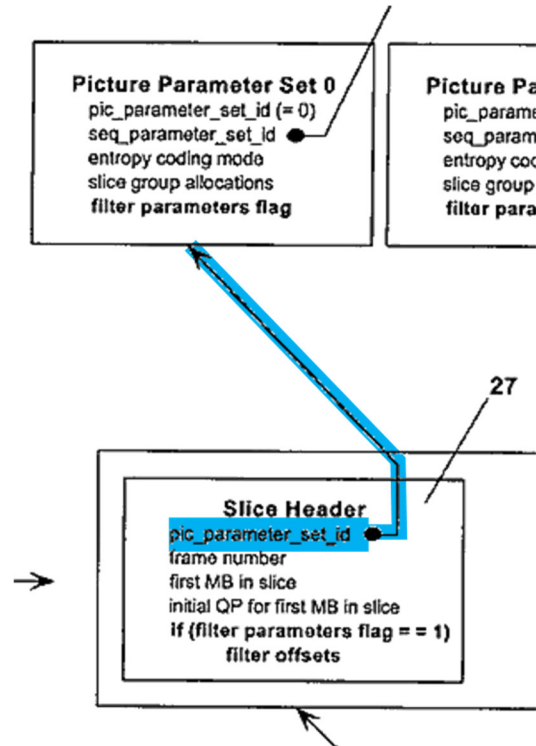
e. 6[d]

Au discloses or suggests 6[d]. *Id.*, ¶136. Au discloses the information in the slice headers “pertain[s] to the decoding of the slice data that follow the slice header

27.” Au, 8:40-43. Au discloses using a frame number in decoding video sequences because the frame number value is a unique identifier for each picture. Saber, ¶136. A decoder uses the frame number in the slice header to know which frame the slice belongs to and is currently being decoded in the picture buffer. *Id.* A POSITA would have understood this to disclose using each piece of information in the slice header, including the frame number, identified above as the “second picture parameter value” limitation, in decoding. *Id.*

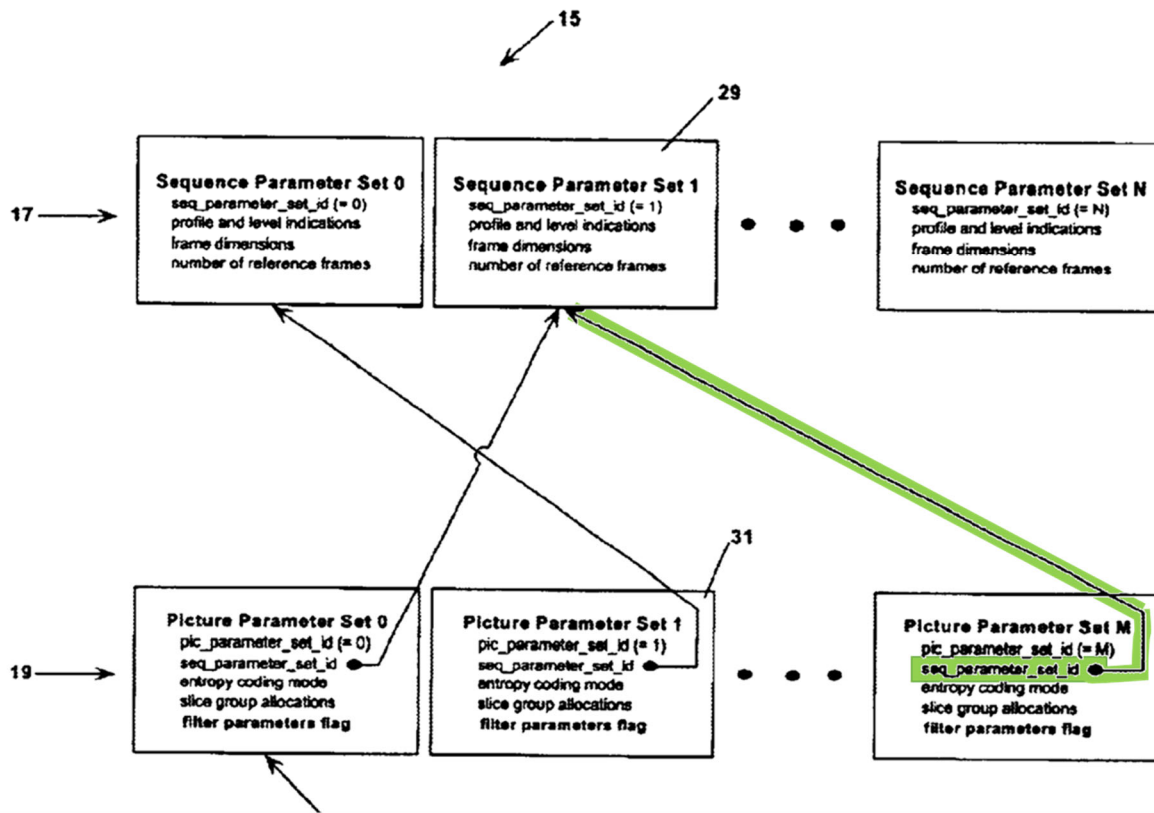
3. Claim 7

Section VII.C.2 explains Au discloses or suggests the “method according to claim 6.” As discussed in Section VII.C.2.d (6[c]), Au discloses a data element in the slice header “contains a reference to a valid picture parameter set **31**, which specifies the picture parameter values...to be used when decoding.” Au, 8:43-48. This is illustrated in Figure 2 (relevant section reproduced below), which shows the slice header includes “pic_parameter_set_id” with an arrow pointing to a picture parameter set. Saber, ¶¶137-38.



4. Claim 8

Section VII.C.2 explains Au discloses or suggests the “method according to claim 6.” As discussed in Section VII.C.2.c (6[b]), Au discloses “[e]ach picture parameter set **31**...refers to the unique ID of a valid sequence parameter set **29**, which selects the active sequence parameters that are used when decoding coded pictures **22** that use the particular picture parameter set [**31**].” Au, 8:30-34. This is illustrated in Figure 2 (relevant section below, annotated), which shows each picture parameter set includes “seq_parameter_set_id” with an arrow pointing to a sequence parameter set. Saber, ¶¶139-40.



5. Claim 9

Section VII.C.2 explains Au discloses or suggests the “method according to claim 6.” As discussed in Section VII.C.2.d (6[c]), Au’s frame number teaches the recited “second picture parameter value.” The “frame number” is a parameter that identifies the appropriate frame number in a given sequence and so is “selected from a group of information of a picture order count and a frame number.” *Id.*, ¶141.

6. Claim 11

a. 11[pre]

If limiting, Au discloses or suggests 11[pre]. *Id.*, ¶142. As discussed in Section VII.C.2.a (6[pre]), Au discloses or suggests “decoding sequences of pictures from a bitstream, wherein parameters are defined in a parameter set and each picture comprises information of one or more slices.” Au further discloses such decoding is done by a decoder. Au, 6:54-55 (“FIG. 3 shows further details of a decoder of the system of FIG. 1”), 10:41-46 (“The decoder **20** (see FIG. **3**) of the receiving participant A,B processes the received bit stream 15 and then reconstructs the coded frame....”), Fig. 3; Saber, ¶142.

b. 11[a]

Au discloses or suggests 11[a]. Saber, ¶143. Au discloses “the processors **16** coordinate the display of successive frames **22** on the monitors **13**, as the video data **12** is communicated between the participants A, B during a video conference.” Au, 7:11-14. This is illustrated in Figure 1 below.

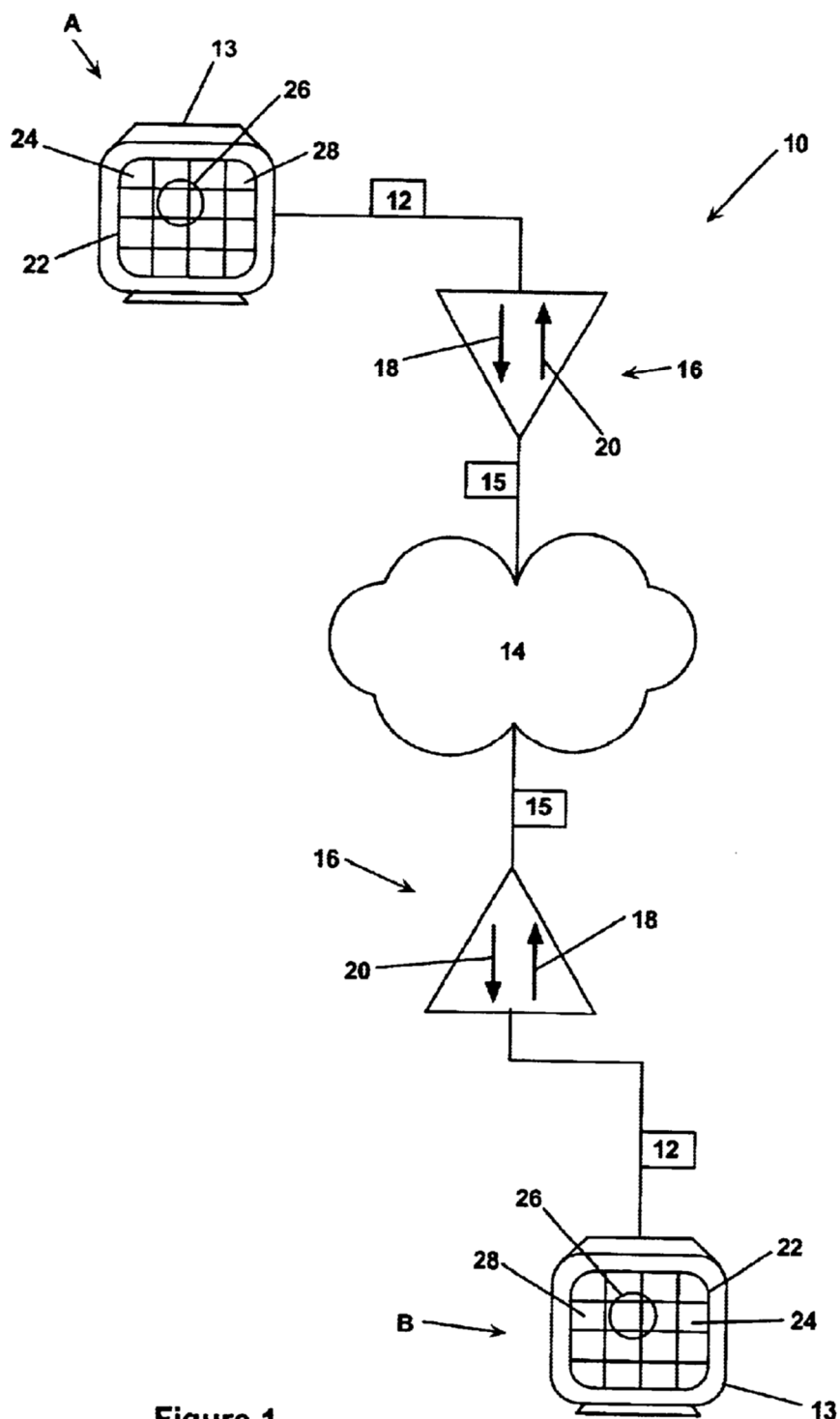


Figure 1

Au's decoder 20 comprises a processor 16. Saber, ¶143. If this limitation requires a processor to be dedicated to decoding, Dr. Saber explains it would have been obvious to a POSITA to implement a decoder in a dedicated processor. *Id.*

c. 11[b]

11[b] is substantially similar to 6[a] (Section VII.C.2.b) except that 11[b] recites “recognizing a sequence parameter set for a sequence of pictures [and] using the parameter set for a picture.” *Id.*, ¶144. Au also teaches these 11[b] additions because Au discloses a sequence parameter set “includes values of parameters that will remain unchanged for an entire video sequence, or from one instantaneous decoder refresh (IDR) picture to the next.” Au, 8:12-15. A POSITA would have understood both “an entire video sequence” and a sequence “from one [IDR] picture to the next” would each be a “sequence of pictures” and thus the sequence parameter set is “for a sequence of pictures,” as 11[b] recites. Saber, ¶144. A POSITA would have also understood at least some of the parameters in Au's sequence parameter set, including frame dimensions would be “for the picture,” as required by 11[b] because that parameter is used for displaying pictures in the correct dimensions. *Id.*; Au, 8:16-19, Fig. 2.

d. 11[c]

Au discloses or suggests 11[c] for substantially the same reasons as 6[b]. Section VII.C.2.c; Saber, ¶145.

e. 11[d]

Au discloses or suggests 11[d] for substantially the same reasons as 6[c].

Section VII.C.2.d; Saber, ¶146.

f. 11[e]

Au discloses or suggests 11[e] for substantially the same reasons as 6[d].

Section VII.C.2.e; Saber, ¶147.

7. Claim 15

a. 15[pre]

If limiting, Au discloses or suggests 15[pre]. Saber, ¶148. Beyond 11[pre] (Section VII.C.6.a), 15[pre] adds only a “receiving device” comprising 11[pre]’s decoder. Au teaches this additional requirement because it discloses the exchange of video data, including decoding, in devices such as “personal computers, wireless devices, surveillance systems, video conferencing systems and set-top boxes.” Au, 7:17-22. A POSITA would have understood those devices to be “receiving devices” that comprise a decoder because such devices receive and store data, and a POSITA would have also understood decoding in Au is performed in such devices because the received data is what is decoded. Saber, ¶148.

b. 15[a]

Au discloses or suggests 15[a] for substantially the same reasons as 11[a].

Section VII.C.6.b; Saber, ¶149.

c. 15[b]

Au discloses or suggests 15[b] for substantially the same reasons as 11[b].

Section VII.C.6.c; Saber, ¶150.

d. 15[c]

Au discloses or suggests 15[c] for substantially the same reasons as 11[c].

Section VII.C.6.d; Saber, ¶151.

e. 15[d]

Au discloses or suggests 15[d] for substantially the same reasons as 11[d].

Section VII.C.6.e; Saber, ¶152.

f. 15[e]

Au discloses or suggests 15[e] for substantially the same reasons as 11[e].

Section VII.C.6.f; Saber, ¶153.

8. Claim 21

a. 21[pre]

If limiting, Au discloses or suggests 21[pre]. Saber, ¶¶154-55. Beyond 6[pre] (Section VII.C.2.a), 21[pre] adds only a “computer readable storage medium embedded with a computer program comprising programming code for execution on a processor” and that program is for the “decoding” similarly recited in 6[pre]. Au discloses this.

As discussed in Section VII.C.6.a (11[pre]), Au’s decoding is done by a decoder. Au, 6:54-55, 10:41-46, Fig. 3; Saber, ¶155. Au further discloses “the

processors **16** coordinate the display of successive frames **22** on the monitors **13**, as the video data **12** is communicated between the participants A, B during a video conference.” Au, 7:11-14. And as discussed in Section VII.C.6.b (11[a]), a POSITA would have understood Au discloses a processor for decoding video. *Id.*; Saber, ¶155. A POSITA would have further understood such a decoder using a processor would have been implemented with programming code in a computer program stored on a computer readable storage medium. Saber, ¶155.

b. 21[a]

Au discloses or suggests 21[a] for substantially the same reasons as 11[b].
Section VII.C.6.c; Saber, ¶156.

c. 21[b]

Au discloses or suggests 21[b] for substantially the same reasons as 6[b].
Section VII.C.2.c; Saber, ¶157.

d. 21[c]

Au discloses or suggests 21[c] for substantially the same reasons as 6[c].
Section VII.C.2.d; Saber, ¶158.

e. 21[d]

Au discloses or suggests 21[d] for substantially the same reasons as 6[d].
Section VII.C.2.e; Saber, ¶159.

9. Claim 23

a. 23[pre]

If limiting, Au discloses or suggests 23[pre] for substantially the same reasons as 11[pre]. Section VII.C.6.a; Saber, ¶160.

b. 23[a]

Au discloses or suggests 23[a] regardless of whether it is construed as MPF. Saber, ¶¶161-64.

Au teaches the function as discussed in Section VII.C.6.c (11[b]). *Id.*, ¶162.

If this term is construed as MPF, Au discloses or suggests the structure discussed in Section VI.A (“a decoder as described in the ’818 patent specification in figure 4 and equivalents thereof”). *Id.*, ¶163. As shown below, the ’818 patent’s Figure 4 discloses a structure including a processor, decoding buffer, program code (decoding buffer and program code both stored in memory).

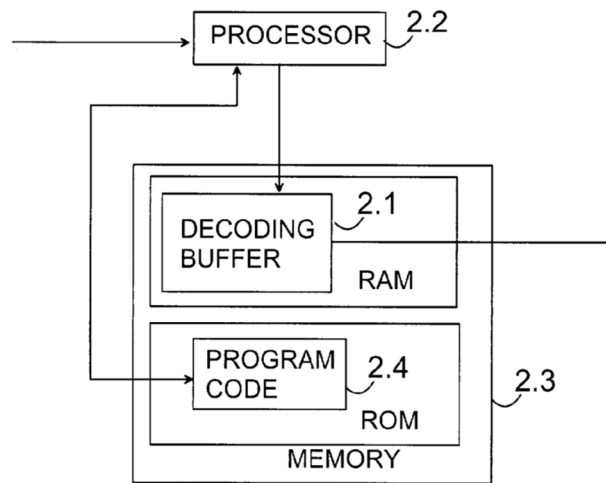


Fig. 4

As discussed in Section VII.C.6.c (11[b]), Au discloses “recognizing a sequence parameter set for a sequence of pictures and forming at least one sequence parameter pertaining to a sequence using the parameter set for a picture.” Similarly, as noted for 6[a], Au discloses this recognition is done “in a decoder.” Au further discloses a decoder for converting a bitstream into a series of coded frames. Au, 6:54-55, 10:41-11:25, Fig. 3. As Section VII.C.6.b (11[a]) explains, Au discloses the decoder includes a processor. As discussed in Section VII.C.8.a (21[pre]), a POSITA would have understood, or found it obvious, that Au’s decoder uses a processor implemented with programming code in a computer program stored on a

computer readable storage medium. Au's decoder also includes a "buffer **500** at the input to the decoder **20**." *Id.*, 11:5-10, 12:20 ("decoder buffer **500**"). Au further discloses the decoder buffer is stored using memory. *Id.*, 12:20-24 (discussing how decoder buffer affects "memory requirements"). For these reasons, a POSITA would have understood Au discloses a decoder with structure the same as, or equivalent to, that of '818 patent Figure 4, and that decoder performs 23[a]'s recited function. Saber, ¶164.

c. 23[b]

Au discloses or suggests 23[b] regardless of whether it is construed as MPF. *Id.*, ¶¶165-67.

Au teaches the function as discussed in Section VII.C.2.c (6[b]). *Id.*, ¶166.

If this term is construed as MPF, Au's decoder discloses or suggests the structure discussed in Section VI.A ("a decoder as described in the '818 patent specification in figure 4 and equivalents thereof") for the same reasons discussed in 23[a] above and that decoder performs 23[b]'s recited function. *Id.*, ¶167.

d. 23[c]

Au discloses or suggests 23[c] regardless of whether it is construed as MPF. *Id.*, ¶¶168-70.

Au teaches the function as discussed in Section VII.C.2.d (6[c]). *Id.*, ¶169.

If this term is construed as MPF, Au's decoder discloses or suggests the structure discussed in Section VI.A ("a decoder as described in the '818 patent specification in figure 4 and equivalents thereof") for the same reasons discussed in 23[a] above and that decoder performs 23[c]'s recited function. *Id.*, ¶170.

e. 23[d]

Au discloses or suggests 23[d] regardless of whether it is construed as MPF. *Id.*, ¶¶171-73.

Au teaches the function as discussed in Section VII.C.2.e (6[d]). *Id.*, ¶172

If this term is construed as MPF, Au's decoder discloses or suggests the structure discussed in Section VI.A ("a decoder as described in the '818 patent specification in figure 4 and equivalents thereof") for the same reasons discussed in 23[a] above and that decoder performs 23[d]'s recited function. *Id.*, ¶173.

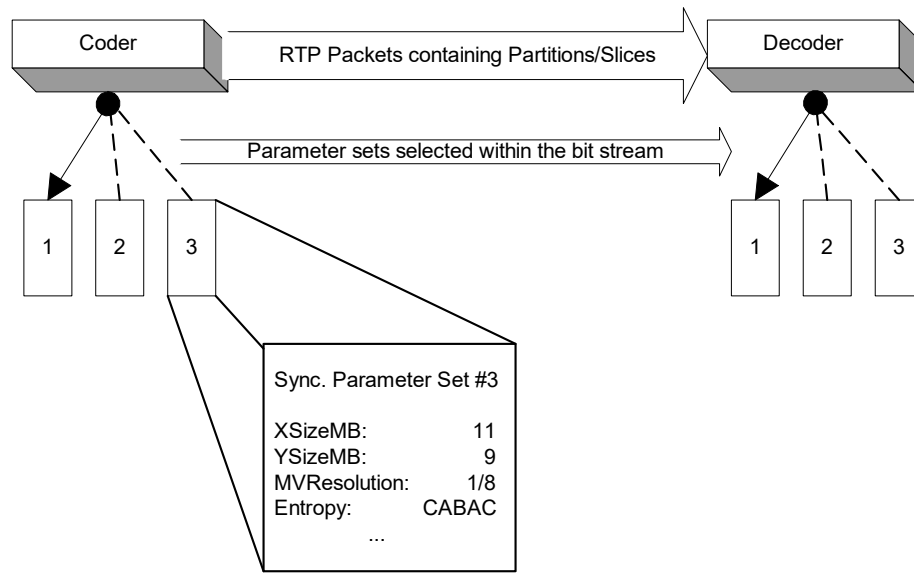
D. Ground 4: Obviousness over VCEG-N52 in view of Sriram

1. VCEG-N52

VCEG-N52 qualifies as a printed publication, which the '818 applicant submitted during prosecution and admitted was prior art. EX-1002, 146. In particular, the '818 applicant included a publication date of "September, 2001" in the Information Disclosure Statement identifying VCEG-N52. *Id.*; *see also* 37 C.F.R. §1.98(b)(5) (identification of publications includes "date, and place of publication"). VCEG-N52 was publicly available before the earliest priority date on

the '818 patent's face. Munford, ¶¶18-30; Reader, ¶¶51-55. Both Ms. Munford and Dr. Reader provide supporting testimony and evidence, showing the printed publication status of VCEG-N52. Munford, ¶18 (reference "Generated: 18 September '01," for a September 2001 meeting in Santa Barbara, California), 19 (Metadata indicates content last saved September 20, 2001), 20-23 (describing process for obtaining document), 24-27 (articles citing VCEG-N52 as published in September 2001); Reader, ¶¶52-53 (reference would have been available for download by the start of the September 2001 meeting, and those interested would have likely reviewed documents on the ITU-T website and would have had access to the reference). Therefore, VCEG-N52 was publicly available in September 2001—and at least by April 2002—and is prior art under §102(b). Munford, ¶30; Reader, ¶¶53-55; Saber, ¶175.

VCEG-N52 describes a "concept for the transport of H.26L over IP and H.324 networks" using parameter sets. VCEG-N52, 1; Saber, ¶176. A video coder generates parameter sets and slice headers that indicate the parameter set to be used. VCEG-N52, 3-4. A video decoder then "maintains a number of parameter set storages" and uses the information from the slice header to determine which parameter set to use for that slice. *Id.*, 4. VCEG-N52 includes a figure, below, showing the transmission of parameter sets within the bit stream. *Id.*, 3.



VCEG-N52 discloses a syntax for transmitting parameter sets, with a parameter set identifier (“ParameterSetNumber”), parameter name (“ParameterName”), and parameter value (“ParameterValue”). *Id.*, 10; Saber, ¶177.

VCEG-N52 discloses different parameters, which are related to different levels of the video hierarchy (Sequence, Group of Pictures (“GOP”), Picture, Slice). VCEG-N52, 8. VCEG-N52 explicitly discloses storing a parameter set identifier in the slice header. *Id.*, 6 (“ParameterSet”); Saber, ¶178.

2. Sriram

Sriram discloses a video decoding system with an explicit “design goal” of “efficiency.” Sriram, 1:43; Saber, ¶179. In particular, Sriram takes advantage of the fact that there are “shared parameters in a bit stream.” Sriram, 1:43-47. Sriram includes an example where “slice headers” can be used to “identify portions of a video frame which can be independently processed.” *Id.*, 1:47-49. As with VCEG-

N52's disclosure of different levels of a video hierarchy, with different parameters at each level, Sriram discloses "several parameters which [are] extracted from the picture, sequence and GOP headers" that are used for decoding a macroblock. *Id.*, 12:64-66, 6:1-5. Sriram defines "[s]everal smaller data structures" containing these parameters, where pointers to these data structures are included in the slice and macroblock structures. *Id.*, 12:66-13:1, 6:19-22. Sriram states as a constraint of the invention, "[a]ll parameter data that is needed at a certain level shall be accessible from the data structure defined for that level." *Id.*, 6:8-11. For example, Sriram discloses a picture parameter ("Picture Horizontal Size"), which is stored in a picture parameter data structure with a pointer to it from every slice data structure. *Id.*, 13:2-9. Sriram provides an exemplary data structure ("PicParam") that "contains all the different picture-level parameters that are also needed at the [macroblock] level." *Id.*, 14:35-54.

Sriram also discloses a "slice data structure," which contains a pointer (indicated with an asterisk) ("picparam") to a PicParam data structure. *Id.*, 15:57-65; Saber, ¶180.

Sriram's slice data structure also includes a pointer to a picture frame data structure ("PicFrame *PicFramePtr"). Sriram, 15:57-65; Saber, ¶181. Because Sriram includes parameters within data structures for each level, a POSITA would have understood (or found it obvious) that Sriram includes sequence parameters in

a sequence parameter data structure, and includes GOP parameters in a GOP parameter data structure. Sriram, 6:8-11; Saber, ¶181. Further, because Sriram’s “primary data structures” have pointers to “secondary data structures,” a POSITA would have understood (or found it obvious) the picture parameter data structure includes a pointer to the GOP parameter data structure, and the GOP parameter data structure includes a pointer to the sequence parameter data structure. Sriram, 6:19-22; Saber, ¶181.

3. Motivation to Combine VCEG-N52 and Sriram

A POSITA would have been motivated to combine VCEG-N52 and Sriram to extend VCEG-N52 based on Sriram’s teachings about efficient data structures and would have had a reasonable expectation of success in doing so for several reasons. Saber, ¶¶182-87.

First, both references disclose video decoding and processing parameters for video decoding. For example, both references disclose (1) the same video hierarchy of sequence, GOP, picture, slices, and macroblocks; VCEG-N52, 4, 8; Sriram, 3:14-18, 6:4-5, 9:29-30; Saber, ¶182; (2) how to efficiently process the parameters for these levels contained within a bitstream; VCEG-N52, 8; Sriram, 1:43-44; Saber, ¶182; (3) storing parameters together, “parameter set” in VCEG-N52 and data structure in Sriram; VCEG-N52, 3; Sriram, 12:64-13:13, 14:35-54; Saber, ¶182; and (4) identifying the parameter set to use for each slice; VCEG-N52, 6 (slice header

includes “ParameterSet” identifier); Sriram, 15:56-67 (“SliceData” data structure includes “PicParam *picparam”); Saber, ¶182.

Relatedly, a POSITA would have been motivated to improve VCEG-N52’s parameter set concept based on Sriram’s teaching of “[s]everal smaller data structures” for holding the “several parameters” extracted from the “picture, sequence and GOP headers.” VCEG-N52, 3; Sriram, 12:64-13:13; Saber, ¶183. Put differently, a POSITA would have been motivated to break up VCEG-N52’s parameter set into multiple parameter sets for each level of the video hierarchy. Saber, ¶183. Doing so would have contributed to the shared goal of VCEG-N52 and Sriram of increasing bandwidth. Sriram, 6:29-33; VCEG-N52, 8; Saber, ¶183. In the combination, each level of parameters would need to be transmitted only once, as part of the related parameter set, as opposed to being included in every parameter set. Saber, ¶¶182-83.

Extending VCEG-N52’s parameter set concept with Sriram’s multiple parameters would involve only routine engineering tasks that would yield predictable results. Saber, ¶¶184-85. In particular, this combination would have been accomplished using well-known software engineering techniques applied to the disclosures of VCEG-N52 and Sriram, and would have been well within the capabilities of a POSITA to program. *Id.* VCEG-N52’s parameter set syntax includes parameters for each level of the video hierarchy. *Id.*

Separating these parameters into different parameter sets would have been a routine software engineering task with expected results. *Id.*, ¶186. Further, to encapsulate all the parameters used in decoding of a picture in a single data structure, a POSITA would have added the items included in Sriram's PicParam data structure to the picture parameter set. *Id.*; Sriram, 14:35-54. Additionally, because Sriram's "primary data structures" have pointers to "secondary data structures," a POSITA would have included, within each parameter set, an identifier for the next higher parameter set (i.e., picture parameter set includes pointer to GOP parameter set ("GOPParameterSetIdentifier"); GOP parameter set includes pointer to sequence parameter set ("SeqParameterSetIdentifier")). Sriram, 6:19-22; Saber, ¶186. Such a reference between levels of the video hierarchy is also taught by Sriram's slice data structure, which includes a pointer to the picture parameter data structure. Sriram, 15:56-67 ("PicParam *picparam"); Saber, ¶¶186-87.

4. Claim 6

a. 6[pre]

If limiting, combining VCEG-N52 and Sriram teaches 6[pre]. Saber, ¶¶188-90. VCEG-N52 and Sriram each disclose a video decoder. VCEG-N52, 4; Sriram, 4:11-12, Figure 2. A POSITA would have understood the combination would also include a decoder. Saber, ¶188.

As to “decoding sequences of pictures from a bitstream” and “each picture comprises information of one or more slices,” both VCEG-N52 and Sriram disclose decoding pictures comprised of slices from a bitstream. VCEG-N52, 5, 6, 12, 1; Sriram, 6:1-5, 4:59-5:14; Saber, ¶189.

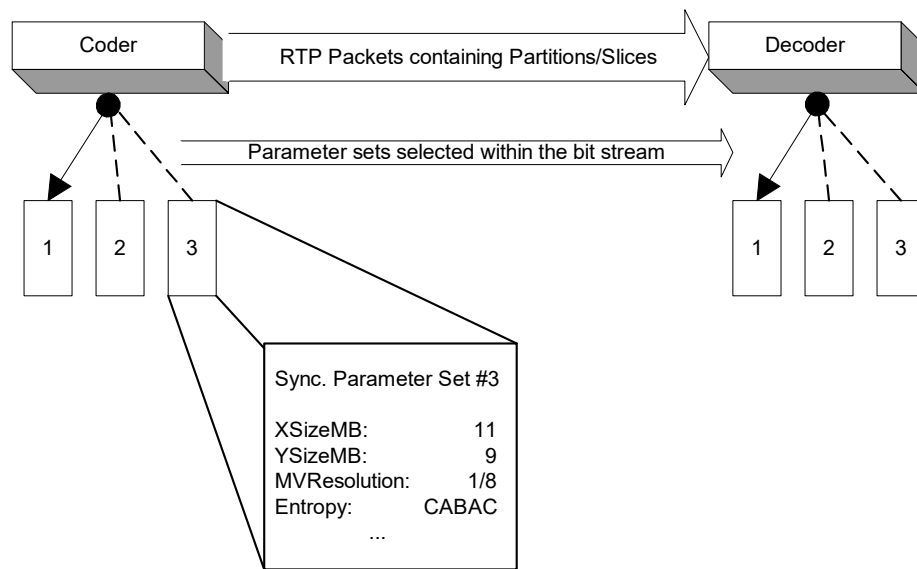
As to “parameters are defined in a parameter set,” as admitted by the ’818 applicant, VCEG-N52 discloses the “Parameter Set Concept.” EX-1001, 2:49-58; EX-1002, 142; VCEG-N52, 3. As discussed in Section VII.D.3, a POSITA would have been motivated to combine VCEG-N52 with Sriram’s teaching of different data structures for different levels of the video hierarchy. Saber, ¶190.

b. 6[a]

Combining VCEG-N52 and Sriram teaches 6[a]. *Id.*, ¶¶191-96. As described above (6[pre]), the ’818 applicant acknowledged VCEG-N52 disclosed the parameter set concept. EX-1001, 2:49-58; EX-1002, 142; VCEG-N52, 3. As further described in Section VII.D.3, a POSITA would have been motivated to extend VCEG-N52’s parameter set concept to each level of the video hierarchy, including the sequence and GOP levels, based on Sriram’s teachings. In particular, Sriram discloses “there are several parameters which were extracted from the picture, *sequence* and *GOP* headers which are needed for processing an individual MB [macroblock]” (Sriram, 12:64-66 (emphasis added)), and “[s]everal smaller data structures are defined and a pointer to these data structures is included in the Slice

and MB data structures” (*id.*, 12:66-13:1). Sriram’s example picture parameter data structure is non-limiting (*id.*, 13:2-9; 17:55-60), and a POSITA would have understood (or found it obvious) that Sriram at least suggests a sequence parameter data structure storing sequence parameters and a GOP parameter data structure storing GOP parameters, including at least VCEG-N52’s “Sequence Parameters” and “Group of Picture Parameters.” VCEG-N52, 8; Saber, ¶191. A POSITA would have been motivated to make this combination so that sequence and GOP parameters would not need to be repeated in each picture parameter data structure to promote Sriram’s stated goal of reducing “data bandwidth.” Sriram, 6:29-33, 13:11-13; Saber, ¶191.

A POSITA would have understood “recognizing, in the decoder” (6[a]) encompasses a decoder parsing the incoming bitstream and identifying a “sequence parameter set.” Saber, ¶192. VCEG-N52 performs such “recognizing.” VCEG-N52’s Figure 1, below, shows “Parameter sets selected within the bit stream,” sent from the “Coder” to the “Decoder.”



VCEG-N52 describes “Transmission of Parameter sets,” including how the information is formatted for the decoder to recognize the parameter set (e.g., “ParameterSetNumber,” “ParameterName,” and “ParameterValue”). VCEG-N52, 9-11; Saber, ¶193.

As described in Section VII.D.3, a POSITA would have been motivated to use VCEG-N52’s parameter set transmission from the coder, and receipt by the decoder, of the parameters related to a sequence and GOP. Saber, ¶194. The GOP parameter set and sequence parameter set are each “a sequence parameter set” as recited by 6[a]. *Id.*, ¶195 (citing EX-1001, 7:20-21, 7:54-57, 8:45-54, Fig. 1).

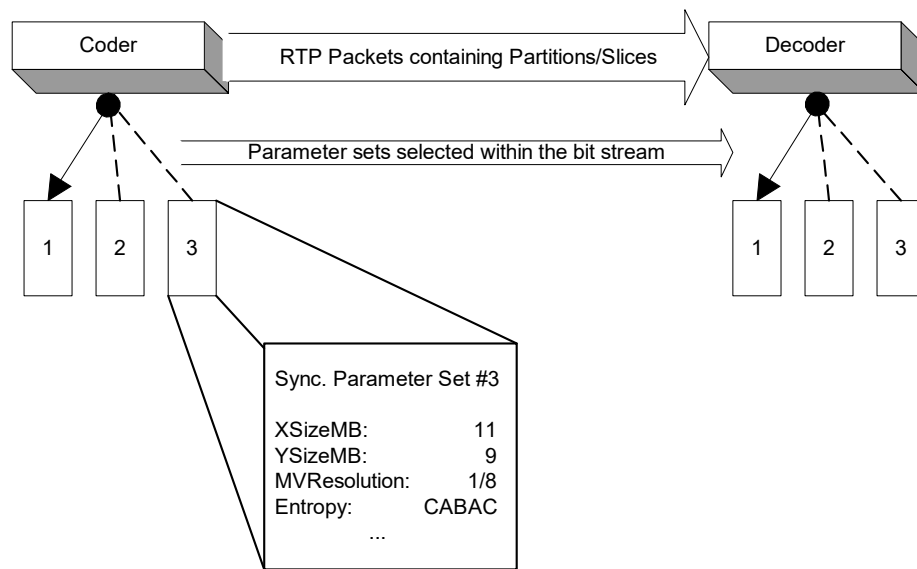
A POSITA would have further understood the combination of VCEG-N52 and Sriram includes “forming” (e.g., creating in local memory) a sequence parameter. Saber, ¶¶195-96. In particular, Dr. Saber provides a syntax for the

combination of VCEG-N52 and Sriram including variables (“SeqParameterValue” and “GOPParameterValue”) containing values of a sequence parameter included in the sequence parameter set and GOP parameter set, respectively. *Id.* A POSITA would have understood these variables and their associated values would be extracted by the decoder and stored in local memory for later use. *Id.*

c. 6[b]

Combining VCEG-N52 and Sriram teaches 6[b]. *Id.*, ¶¶197-202. VCEG-N52 discloses the parameter set concept, and Sriram discloses a picture parameter data structure. VCEG-N52, 3; Sriram, 15:57-65. Sriram’s picture parameter data structure includes multiple picture parameters, each of which teaches “at least one first picture parameter value,” including, *inter alia*, “horizontal_size,” “vertical_size,” and “chroma_format.” Sriram, 14:35-54. As discussed in Section VII.D.3, a POSITA would have been motivated to combine VCEG-N52 with the picture parameters stored in Sriram’s picture parameter data structure. Saber, ¶197.

A POSITA would have understood “recognizing, in the decoder” (6[b]) encompasses a decoder parsing the incoming bitstream and identifying a “picture parameter set.” *Id.*, ¶198. VCEG-N52 performs such “recognizing.” VCEG-N52’s Figure 1, below, shows “Parameter sets selected within the bit stream,” sent from the “Coder” to the “Decoder.”



VCEG-N52 also describes “Transmission of Parameter sets,” including how the information is formatted for the decoder to recognize the parameter set (e.g., “ParameterSetNumber,” “ParameterName,” and “ParameterValue”). VCEG-N52, 9-11; Saber, ¶199.

In the combination of VCEG-N52 and Sriram, the picture parameter set would include the parameters stored in Sriram’s picture parameter data structure (“PicParam”). Sriram, 14:35-54; Saber, ¶200.

Further, as described in Section VII.D.3, because Sriram’s “primary data structures” have pointers to “secondary data structures,” a POSITA would have included, within the picture parameter set, a pointer to the sequence parameter set. Sriram, 6:19-22; Saber, ¶201.

A POSITA would have further understood the combination of VCEG-N52 and Sriram includes “forming” (e.g., creating in local memory) a picture parameter.

Saber, ¶202. In particular, “PicParameterValue” is a variable containing the value of a picture parameter included in the picture parameter set, and a POSITA would have understood that such a variable and its associated value would be extracted by the decoder and stored in local memory for later use. *Id.*

d. 6[c]

Combining VCEG-N52 and Sriram teaches 6[c]. Saber, ¶¶203-05. Both VCEG-N52 and Sriram disclose sending slice headers from the coder to the decoder, and extracting information from such slice headers. VCEG-N52, 4; Sriram, 1:47-49, 9:26-29.

A POSITA would have understood Sriram’s slice data structure (“SliceData”) includes both slice data and slice header information (e.g., a pointer to a data bit stream (“btstream”), as well as variables that relate to the data (e.g., “VideoType,” “PicFramePtr”)). Sriram, 15:9-10, 15:52-66; Saber, ¶204. Although Sriram does not provide source code for “PicFrame” structure to which PicFramePtr points, a POSITA would have understood from Sriram’s disclosure what such a structure would be. Saber, ¶204. For example, Sriram discloses structures written in the C programming language, using the “typedef struct” keywords. *E.g.*, Sriram, 13:34-47, 14:26-34, 14:40-53, 14:63-15:7; Saber, ¶204. A POSITA would have understood the PicFrame would also have been defined using “typedef struct.” Saber, ¶204. Using standard programming techniques, a POSITA would have

included in such a structure the proper fields for all the data describing the Picture/Frame. *Id.* As Sriram states, this would have included “the starting address of all the frames that are needed” for decoding. Sriram, 15:9-10; Saber, ¶204. A POSITA would have understood a proper reference to the frame being decoded would be included in “all the frames that are needed” for decoding because (1) a POSITA would need to know where to place the decoded data and (2) data from the frame being decoded may be needed for INTRA and/or INTER coding. Saber, ¶204. As a further use of data encapsulation, a POSITA would have included the identifier of the frame being decoded (i.e., the “frame number”) in the PicFrame. *Id.* Because each of the addresses of the frame being decoded and the frame number of the frame being decoded are defined by the frame being decoded, a POSITA would have understood these variables are picture parameters, and all of the slice data structures for a particular picture would have the same value for these parameters (i.e., “one second picture parameter value using information of a slice header,” which “remain[s] unchanged at least in all slice headers of one picture.”). *Id.* In VCEG-N52 and Sriram, the slice data structure and the PicFrame data structure would have been stored in the decoder (“forming, in the decoder, at least one second picture parameter value using information of a slice header”). *Id.*; Sriram, 4:59-62, 5:51-6:43.

Alternatively, a POSITA would have been motivated to include the identifier for the frame as a field within the slice data structure. Saber, ¶205. VCEG-N52 discloses a PictureID field, which “represents the PictureID of the picture the slice belongs to” as a variable “traditionally associated with the Slice header.” VCEG-N52, 5. A POSITA would have understood that for non-interlaced video, the majority of digital video, a picture and a frame are interchangeable terms. Saber, ¶205. Similarly, Sriram states its invention is applicable to H.263. Sriram, 5:38-45. The H.263 standard describes a Group of Blocks (GOB) Frame ID, which “shall have the same value in every GOB (or slice) header of a given picture.” EX-1022, 19, 32-33; Saber, ¶205. Given these teachings, in VCEG-N52 and Sriram, a POSITA would have included the frame number (“one second picture parameter value using information of a slice header,” which “remain[s] unchanged at least in all slice headers of one picture”) in the slice data structure, which contains slice header information (“forming, in the decoder, at least one second picture parameter value using information of a slice header”). Saber, ¶205.

e. 6[d]

The combination of VCEG-N52 and Sriram teaches 6[d]. *Id.*, ¶206. As described above, the frame number value, stored in the slice data structure and/or the PicFrame data structure is “at least one second picture parameter value.” A POSITA would have understood, or found it obvious, that the frame number is

used in the decoder, because it indicates to the decoder in which frame to decode each slice. *Id.* Therefore, the value of the frame number is “the at least one second picture parameter value in decoding.”

5. Claim 7

Section VII.D.4 explains the combination of VCEG-N52 and Sriram teaches the “method according to claim 6.” As discussed in Section VII.D.4.d (6[c]), that combination teaches Sriram’s slice data structure includes slice header information and a pointer to a picture parameter data structure. Sriram, 15:52-66 (“PicParam*picparam”); Saber, ¶207. As discussed in Section VII.D.4.c (6[b]), in the combination of VCEG-N52 and Sriram, the picture parameter data structure is a picture parameter set and a POSITA would have understood the “PicParam” variable within Sriram’s slice data structure (“slice header”) to be a reference (e.g., a pointer as in Sriram (“picparam”) or an integer reference of the picture parameter set (“PicParameterSetNumber”)) to the combination’s picture parameter set. Saber, ¶207.

6. Claim 8

Section VII.D.4 explains the combination of VCEG-N52 and Sriram teaches the “method according to claim 6.” As discussed in Sections VII.D.3 and VII.D.4.c (6[b]), in the combination of VCEG-N52 and Sriram, a POSITA would have included, within the picture parameter set, a pointer to the sequence parameter set

because Sriram's "primary data structures" have pointers to "secondary data structures." Sriram, 6:19-22; Saber, ¶208.

"GOPParameterSetIdentifier" is a reference to a GOP parameter set. Saber, ¶209. As discussed in Section VII.D.4.b (6[a]), a POSITA would have understood a GOP parameter set to be a "sequence parameter set." *Id.* Thus, the picture parameter set of the combination of VCEG-N52 and Sriram includes a reference to a sequence parameter set.

If a GOP parameter set is not a "sequence parameter set," the combination still teaches claim 8. *Id.*, ¶210. As discussed in Section VII.D.3, in the combination of VCEG-N52 and Sriram, the parameter set at each level includes a reference to the parameter set at the level above it. *Id.* In other words, each picture parameter set has a reference ("GOPParameterSetIdentifier") to the GOP parameter set above it, and each GOP parameter set has a reference ("SeqParameterSetIdentifier") to the sequence parameter set above it. *Id.* There is, therefore, an indirect reference from every picture parameter set to the sequence parameter set two levels above it. *Id.* A POSITA would have understood claim 8's including "a reference to a sequence parameter set" encompasses such an indirect reference. *Id.*

If a GOP parameter set is not a "sequence parameter set," and claim 8 requires a direct reference from the picture parameter set to a sequence parameter set, a POSITA would have been motivated to modify the combination of VCEG-N52 and

Sriram to include a direct reference. *Id.*, ¶211. Sriram teaches “[h]aving two levels of indirection may not be desirable for some parameters which are extensively used by all the processing blocks.” Sriram, 13:13-15. Accordingly, a POSITA would be motivated to either (1) combine the GOP parameter set and the sequence parameter set (i.e., including GOP parameters and sequence parameters in the same parameter set), or (2) include a direct reference from the picture parameter set to both the GOP parameter set and the sequence parameter set. Saber, ¶211. In either case, the combination would include a direct reference from the picture parameter set to a sequence parameter set. *Id.*

7. Claim 9

Section VII.D.4 explains the combination of VCEG-N52 and Sriram teaches the “method according to claim 6.” As discussed in Sections VII.D.4.d (6[c]) and VII.D.4.e (6[d]), in the combination of VCEG-N52 and Sriram, the “second picture parameter value” is the value of the frame number contained in Sriram’s slice data structure and/or the PicFrame data structure, as required by this claim. Sriram, 15:52-66; Saber, ¶212.

8. Claim 11

a. 11[pre]

If limiting, the combination of VCEG-N52 and Sriram teaches 11[pre]. Saber, ¶213. Beyond 6[pre] (Section VII.D.4.a), both VCEG-N52 and Sriram disclose that

such decoding is done by a decoder. VCEG-N52, 4; Sriram, 4:11-12, Figure 2; Saber, ¶213.

b. 11[a]

The combination of VCEG-N52 and Sriram teaches 11[a]. Saber, ¶¶214-17.

VCEG-N52 discloses a coder and decoder connected via an IP network. VCEG-N52, 1, 2. A POSITA would have understood such a decoder would have been implemented using a processor. Saber, ¶215.

Sriram discloses a central processing unit that reads and executes a number of “executable programs utilized to implement the video decoding processes of the invention.” Sriram, 4:34-58, Figs. 1, 2; Saber, ¶216. These executable programs include a syntax analyzer, monitor processor, sub-processors, synchronous/synchronization processor, and display processor. Saber, ¶216. The syntax analyzer program “extract[s] information from the video bit stream.” Sriram, 5:17-19.

In the combination of VCEG-N52 and Sriram, a POSITA would have understood a processor performs the steps in 11[b]-11[e] as discussed below. Saber, ¶217.

c. 11[b]

11[b] is substantially similar to 6[a] (Section VII.D.4.b) except 11[b] recites “recognizing a sequence parameter set for a sequence of pictures [and] using the

parameter set for a picture.” 11[b]’s additions are taught by the combination of VCEG-N52 and Sriram. Saber, ¶218. Both VCEG-N52 and Sriram disclose a hierarchy of syntax layers, with the level above picture being a group of pictures. VCEG-N52, 8; Sriram, 6:1-7; Saber, ¶218. A POSITA would have understood a group of pictures is a sequence of pictures. Saber, ¶218. Further, a POSITA would have understood the values stored in the GOP parameter set, such as “XsizeMB” and “YSizeMB,” which together provide the dimensions of each picture within the GOP, in macroblocks, would be used for “a picture” (i.e., every picture within the sequence). *Id.*; VCEG-N52, 8.

d. 11[c]

The combination of VCEG-N52 and Sriram teaches 11[c] for substantially the same reasons as 6[b]. Section VII.D.4.c; Saber, ¶219.

e. 11[d]

The combination of VCEG-N52 and Sriram teaches 11[d] for substantially the same reasons as 6[c]. Section VII.D.4.d; Saber, ¶220.

f. 11[e]

The combination of VCEG-N52 and Sriram teaches 11[e] for substantially the same reasons as 6[d]. Section VII.D.4.e; Saber, ¶221.

9. Claim 15

a. 15[pre]

If limiting, the combination of VCEG-N52 and Sriram teaches 15[pre]. Saber, ¶222. 15[pre] adds only a “receiving device” comprising such a decoder to 11[pre] (Section VII.D.8.a). Both VCEG-N52 and Sriram disclose the decoder is included within a “receiving device.” VCEG-N52 discloses the use of “H.324 terminals” and “IP based terminals” that contain a decoder. VCEG-N52, 1, 3; *see also id.*, 11 (referring to “A Videoconferencing terminal”). Similarly, Sriram uses its invention in workstations, and its decoder is part of a system that is connected to input/output devices, including a mouse, keyboard, computer monitor, and printer. Sriram, 3:54-59, 4:34-41, 17:36-39.

b. 15[a]

The combination of VCEG-N52 and Sriram teaches 15[a] for substantially the same reasons as 11[a]. Section VII.D.8.b; Saber, ¶223.

c. 15[b]

The combination of VCEG-N52 and Sriram teaches 15[b] for substantially the same reasons as 11[b]. Section VII.D.8.c; Saber, ¶224.

d. 15[c]

The combination of VCEG-N52 and Sriram teaches 15[c] for substantially the same reasons as 11[c]. Section VII.D.8.d; Saber, ¶225.

e. 15[d]

The combination of VCEG-N52 and Sriram teaches 15[d] for substantially the same reasons as 11[d]. Section VII.D.8.e; Saber, ¶226.

f. 15[e]

The combination of VCEG-N52 and Sriram teaches 15[e] substantially the same reasons as 11[e]. Section VII.D.8.f; Saber, ¶227.

10. Claim 21

a. 21[pre]

If limiting, the combination of VCEG-N52 and Sriram teaches 21[pre]. Saber, ¶¶228-30. Beyond 6[pre] (Section VII.D.8.a) and 11[a] (Section VII.D.8.b), 21[pre] adds only a “computer readable storage medium embedded with a computer program comprising programming code for execution on a processor” and that program is for the “decoding” similarly recited in 6[pre]. The combination of VCEG-N52 and Sriram discloses these additions. *Id.*, ¶¶228-29.

A POSITA would have understood the combination’s decoder using a processor would have been implemented with programming code in a computer program stored on a computer readable storage medium. *Id.*, ¶230. Sriram describes “system memory” (a computer readable storage medium) that “stores a set of executable programs” (programming code) “utilized to implement the video decoding processes of the invention.” Sriram, 4:34-46. That system memory is

connected to the central processing unit via a system bus. *Id.*, 4:34-38, Fig. 1. A POSITA would have understood the connection between the central processing unit and the system memory would be so the central processing unit (a processor) could access and execute the programs stored in system memory. Saber, ¶230.

b. 21[a]

The combination of VCEG-N52 and Sriram teaches 21[a] for substantially the same reasons as 11[b]. Section VII.D.8.c; Saber, ¶231.

c. 21[b]

The combination of VCEG-N52 and Sriram teaches 21[b] for substantially the same reasons as 6[b]. Section VII.D.4.c; Saber, ¶232.

d. 21[c]

The combination of VCEG-N52 and Sriram teaches 21[c] for substantially the same reasons as 6[c]. Section VII.D.4.d; Saber, ¶233.

e. 21[d]

The combination of VCEG-N52 and Sriram teaches 21[d] for substantially the same reasons as 6[d]. Section VII.D.4.e; Saber, ¶234.

11. Claim 23

a. 23[pre]

If limiting, the combination of VCEG-N52 and Sriram teaches 23[pre] for substantially the same reasons as 11[pre]. Section VII.D.8.a; Saber, ¶235.

b. 23[a]

The combination of VCEG-N52 and Sriram teaches 23[a] regardless of whether it is construed as MPF. Saber, ¶¶236-38.

The combination of VCEG-N52 and Sriram teaches the function as discussed in Section VII.D.8.c (11[b]). *Id.*, ¶237.

If 23[a] is construed as MPF, the combination of VCEG-N52 and Sriram teaches the structure discussed in Section VI.A(“a decoder as described in the ’818 patent specification in figure 4 and equivalents thereof”). Saber, ¶238. As shown below, ’818 patent Figure 4 discloses a structure including a processor, decoding buffer, program code (decoding buffer and program code both stored in memory).

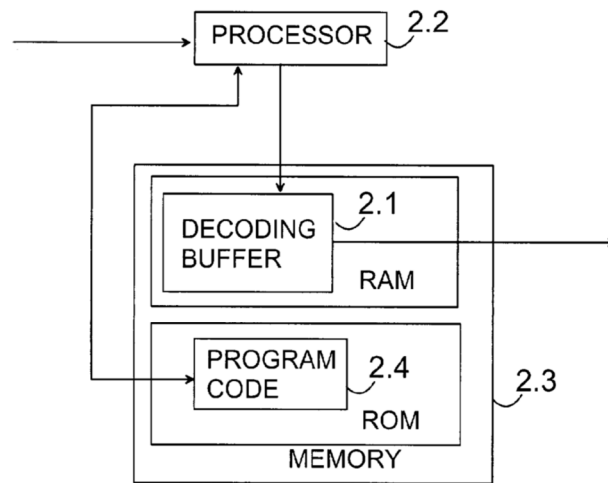


Fig. 4

As discussed in Section VII.D.8.c (11[b]), the combination of VCEG-N52 and Sriram teaches its decoding is performed by a decoder. VCEG-N52, 1, 2; Sriram, 4:34-58, Figs. 1, 2; Saber, ¶239. Section VII.D.8.b (11[a]) explains this combination teaches a processor configured to perform the recited steps that follow. Section VII.D.10.a (21[pre]) explains a POSITA would have understood, or found it obvious, that the combination uses a processor implemented with programming code in a computer program stored on a computer readable storage medium. Saber, ¶239. In particular, Sriram discloses storing executable programs in system memory for decoding. Sriram, 4:34-46. The combination of VCEG-N52 and Sriram also teaches

using a decoding buffer. VCEG-N52 discloses a “playout buffer” that is filled with packets received by the decoder. VCEG-N52, 7. Sriram discloses a “video buffer,” which contains the “frame outputs” from the “synchronization processor.” Sriram, 4:54-58, 5:9-14, Fig. 1. Sriram further discloses “video frames” that are “buffered in memory.” *Id.*, 2:52-55, Fig. 2 (“Buffered Video Frames”). Therefore, a POSITA would have understood the combination of VCEG-N52 and Sriram teaches a decoder with structure the same as, or equivalent to, that of ’818 Figure 4, and that decoder performs 23[a]’s recited function. Saber, ¶239.

c. 23[b]

The combination of VCEG-N52 and Sriram teaches 23[b] regardless of whether it is construed as MPF. *Id.*, ¶¶240-42.

The combination of VCEG-N52 and Sriram teaches the function as discussed in Section VII.D.4.c (6[b]). *Id.*, ¶241.

If 23[b] is construed as MPF, the combination of VCEG-N52 and Sriram teaches the decoder structure discussed in Section VI.A (“a decoder as described in the ’818 patent specification in figure 4 and equivalents thereof”) for the same reasons discussed in 23[a] above and that decoder performs 23[b]’s recited function. Saber, ¶242.

d. 23[c]

The combination of VCEG-N52 and Sriram teaches 23[c] regardless of whether it is construed as MPF. *Id.*, ¶¶243-45.

The combination of VCEG-N52 and Sriram teaches the function as discussed in Section VII.D.4.d (6[c]). *Id.*, ¶244.

If 23[c] is construed as MPF, the combination of VCEG-N52 and Sriram teaches the decoder structure discussed in Section VI.A (“a decoder as described in the ’818 patent specification in figure 4 and equivalents thereof”) for the same reasons discussed in 23[a] above and that decoder performs 23[c]’s recited function. Saber, ¶245.

e. 23[d]

The combination of VCEG-N52 and Sriram teaches 23[d] regardless of whether it is construed as MPF. *Id.*, ¶¶246-48.

The combination of VCEG-N52 and Sriram teaches the function as discussed in Section VII.D.4.e (6[d]). *Id.*, ¶247.

If 23[d] is construed as MPF, the combination of VCEG-N52 and Sriram teaches the decoder structure discussed in Section VI.A (“a decoder as described in the ’818 patent specification in figure 4 and equivalents thereof”) for the same reasons discussed in 23[a] above and that decoder performs 23[d]’s recited function. Saber, ¶248.

VIII. Discretionary Factors Favor Institution

A. *Fintiv*

The Board should not deny institution under *Fintiv*. Three concurrent proceedings involve the '818 patent, two before the district court, and one before the ITC. *Fintiv*, however, does not apply to ITC proceedings. K. Vidal, Memorandum, U.S. PAT. AND TRADEMARK OFFICE, 2-3 (June 21, 2022).

For the district court proceedings, the relevant *Fintiv* factors weigh heavily against discretionary denial. Those proceedings were initiated in November 2023, *Nokia Technologies Oy v. Amazon.com, Inc.*, Case No. 1:23-cv-01236, ECF No. 6, (D. Del. Nov. 3, 2023) and *Nokia Corporation et al. v. Amazon.com, Inc.*, Case No. 1:23-cv-01232, ECF No. 8, (D. Del. Nov. 3, 2023), and no schedule has been entered. Any trial would, therefore, be expected well after the FWD deadline (likely September 2025). Further, it is unclear whether any overlap yet exists between this Petition and issues that might be raised in the district court. So, discretionary denial under *Fintiv* is unwarranted.

B. §325(d)

The Board should likewise not exercise its discretion under §325(d) to deny institution. *See Becton, Dickinson & Co. v. B. Braun Melsungen AG*, IPR2017-01586, Paper 8, 17-18 (Dec. 15, 2017) (precedential); *Advanced Bionics, LLC v.*

MED-EL Elektromedizinische Geräte GmbH, IPR2019-01469, Paper 6, 9-11 (Feb. 13, 2020) (precedential).

Considering the first part of the *Advanced Bionics* framework, including *Becton, Dickinson* factors (a), (b), and (d), although some of the same references were before the Office, that art was never discussed by the Examiner or applied to the claims. Although Au was cited in an Information Disclosure Statement and listed on the face of the patent, the Examiner never relied on it during prosecution. *See GMG Prods. LLC v. Traeger Pellet Grills LLC*, PGR2019-00024, Paper 17, 27 (July 17, 2019) (finding listing on an IDS alone “does not favor denying institution”). Petitioner acknowledges the Examiner applied Joch during prosecution, and although there is no priority claim between Joch and Au, the references include some overlapping disclosure. Nevertheless, the Examiner cited Joch as a secondary reference during prosecution and did not rely on the same disclosures for which Au is relied upon in this Petition. In particular, although the Examiner relied on “pic_parameter_set_id,” in Joch’s Figure 2 (EX-1002, 392), Petitioner relies on “frame_num” in Au’s Figure 2. Thus, that Au and Joch have some overlapping disclosure does not show the Examiner considered the same or substantially the same art or arguments. In addition, JVT-E022d7 and Linzer incorporating it were never cited during prosecution. And although VCEG-N52 was considered during prosecution, it was not considered in combination with Sriram. Accordingly, the

same or substantially the same art or arguments were not previously presented to or considered by the Office; there is no need to go to *Advanced Bionics* part two.

Because some of the art was considered by the Examiner, or PO may argue some was, Petitioner discusses the second part of the *Advanced Bionics* framework. If the Board finds Au is substantially similar to Joch, the Examiner erred by overlooking specific teachings of Joch, which are also included in Au. In particular, although the Examiner focused on Joch's "pic_parameter_set_id" in the slice header as teaching a "picture parameter" (EX-1002, 392), the Examiner failed to recognize Joch's "frame_num" (EX-1020, Fig. 2) teaches a picture parameter. Such an application of Joch's teachings would have been consistent with PO's infringement contentions in the co-pending ITC investigation, where PO alleges "frame_num" teaches a "second picture parameter value remaining unchanged at least in all slice headers of one picture." EX-1016, 15-16. The Examiner erred in not applying Joch's "frame_num" as a "picture parameter" during prosecution, and in not applying other cumulative teachings from Joch in rejecting the pending claims during prosecution. Thus, if Au is cumulative of Joch, §325(d) should not apply to Ground 3 here because of Examiner error.

Similarly, PO may argue that VCEG-N52 was raised during prosecution and/or is cumulative of Visharam. The Examiner erred, however, by not applying

VCEG-N52 or considering it in combination with Sriram (not before the Examiner).

Thus, §325(d) should not apply to Ground 4.

IX. CONCLUSION

Inter Partes Review of the Challenged Claims is respectfully requested.

X. MANDATORY NOTICES

A. Real Party in Interest

Petitioners identify themselves, as real parties in interest. Out of an abundance of caution, Petitioners also disclose Twitch Interactive, Inc. as a potential real party in interest.

B. Related Matters

To the best of Petitioners' knowledge, the '818 patent has been involved in the following district court and ITC litigation:

- *Nokia Technologies Oy v. Amazon.com, Inc., et al.*, 1-23-cv-01236 (D. Del. Oct. 31, 2023)
- *Nokia Technologies Oy v. HP Inc.*, 1-23-cv-01237 (D. Del. Oct. 31, 2023)
- *Nokia Corporation et al. v. Amazon.com, Inc. et al.*, Case No. 1:23-cv-01232 (D. Del. Oct. 27, 2023).
- *Nokia Technologies Oy v. Lenovo (Shanghai) Electronics Technology Co. Ltd. et al.*, Case No. 5:19-cv-00427 (E.D.N.C. Sept. 25, 2019).

- *Certain Electronic Devices, Including Computers, Tablet Computers, and Components and Modules Thereof*, Inv. No. 337-TA-1208 (ITC Aug. 10. 2020).
- *Certain Video Capable Electronic Devices, Including Computers, Streaming Devices, Televisions, and Components and Modules Thereof*, Inv. No. 337-TA-1380 (ITC Nov. 30. 2023).

C. Notice of Counsel and Service Information

Lead Counsel: Jessica Kaiser, Reg. No. 58,937 PERKINS COIE LLP 1900 Sixteenth Street, Suite 1400 Denver, CO 80202 Tel: (303) 291-2300 Kaiser-ptab@perkinscoie.com	Backup Counsel: Robin Brewer, Reg. No. 60,951 PERKINS COIE LLP 505 Howard Street Suite 1000 San Francisco, CA 94105 Tel: (415) 344-7000 Brewer-ptab@perkinscoie.com
	David Pekarek Krohn, Reg. No. 69,443 PERKINS COIE LLP 33 East Main Street, Suite 201 Madison, WI 53703-3095 Tel: (608) 663-7460 pekarekkrohn-ptab@perkinscoie.com
	R. Tyler Kendrick, Reg. No. 73,930 PERKINS COIE LLP 1201 Third Avenue Suite 4900 Seattle, WA 98101-3099 Tel: (206) 359-8000 kendrick-ptab@perkinscoie.com

	Anni Siitonen, Reg. No. 74,974 PERKINS COIE LLP 1201 Third Avenue Suite 4900 Seattle, WA 98101-3099 Tel: (206) 359-8000 Siitonen_ptab@perkinscoie.com
--	---

Respectfully submitted,

/ Jessica Kaiser /

Jessica Kaiser
Reg. No. 58,937
Attorney for Petitioner

PERKINS COIE LLP
1900 Sixteenth Street, Suite 1400
Denver, CO 80202-5255

Date: April 1, 2024

CERTIFICATE OF WORD COUNT UNDER 37 CFR §42.24(d)

Pursuant to 37 C.F.R. §42.24(a), Petitioner hereby certifies that portions of the above-captioned Petition for *Inter Partes* Review of U.S. Patent 7,724,818, in accordance with and reliance on the word count provided by the word-processing system used to prepare this Petition, that the number of words in this paper is 13,987. Pursuant to 37 C.F.R. §42.24(a), this word count is in compliance and excludes the table of contents, table of authorities, mandatory notices under §42.8, certificate of service, certificate of word count, appendix of exhibits, and any claim listing. This word count was prepared using Microsoft Word.

Respectfully submitted,

/ *Jessica Kaiser* /

Jessica Kaiser
Reg. No. 58,937
Attorney for Petitioner

Date: April 1, 2024
PERKINS COIE LLP
1900 Sixteenth Street, Suite 1400
Denver, CO 80202-5255

CERTIFICATE OF SERVICE

The undersigned hereby certifies that true copies of the Petition for *Inter Partes* Review of U.S. Patent No. 7,724,818 and supporting materials (Exhibits and Power of Attorney) were served via overnight delivery on the Patent Owner at the correspondence address of record as listed on PAIR:

WARE, FRESSOLA, MAGUIRE & BARBER LLP
BRADFORD GREEN, BUILDING 5
755 MAIN STREET
P.O. Box 224
MONROE, CT 06468

A courtesy copy was also sent via electronic mail to Patent Owner's litigation counsel listed below:

Brian E. Farnan
bfarnan@farnanlaw.com
Michael J. Farnan
mfarnan@farnanlaw.com
Warren H. Lipschitz
wlipschitz@mckoolsmith.com
Alexandra F. Easley
aeasley@mckoolsmith.com
Joshua Budwin
jbudwin@mckoolsmith.com
R. Mitch Verboncoeur
mverboncoeur@mckoolsmith.com
Josh Newcomer
jnewcomer@mckoolsmith.com
Kevin Burgess
kburgess@mckoolsmith.com
Theodore Stevenson, III
ted.stevenson@alston.com
John D. Haynes

john.haynes@alston.com
Nicholas T. Tsui
nick.tsui@alston.com
Shawn Gannon
shawn.gannon@alston.com

Respectfully submitted,

/ Jessica Kaiser /

Jessica Kaiser
Reg. No. 58,937
Attorney for Petitioner

Date: April 1, 2024