Filed: July 23, 2025

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SNAP INC.,

Petitioner

v.

NOKIA TECHNOLOGIES OY,

Patent Owner.

IPR2025-01338
U.S. Patent No. 7,724,818

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

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

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	Intentionally Left Blank
1006	Intentionally Left Blank
1007	Intentionally Left Blank
1008	U.S. Patent No. 6,646,578 (“Au”)
1009	Intentionally Left Blank
1010	Intentionally Left Blank
1011	Intentionally Left Blank
1012	H.26L over IP and H.324 Framework (“VCEG-N52”), available at https://www.itu.int/wftp3/av-arch/videosite/0109_San/VCEG-N52.doc
1013	Analysis of fractional pixel interpolation for MC prediction (“VCEG-N55”), available at https://www.itu.int/wftp3/avarch/video-site/0109_San/VCEG-N55.doc.
1014	Declaration of Sylvia Hall-Ellis (“Hall-Ellis”)
1015	A. Murat Tekalp, DIGITAL VIDEO PROCESSING (Prentice Hall pub., 1st ed. 1995) (“Tekalp”)
1016	Intentionally Left Blank
1017	Intentionally Left Blank
1018	Intentionally Left Blank
1019	Intentionally Left Blank
1020	U.S. Patent No. 7,227,901 (“Joch”)
1021	U.S. Patent No. 6,539,059 (“Sriram”)

Exhibit	Reference
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 (2nd ed. 2004)
1024	Brian W. Kernighan & Dennis M. Ritchie, THE C PROGRAMMING LANGUAGE (2nd ed. 1988)

LISTING OF CHALLENGED CLAIMS

Claim 1	
1[pre]	A method for encoding sequences of pictures into a bitstream, wherein parameters are defined in a parameter set and each picture comprises information of one or more slices, the method comprising:
1[a]	defining, in an encoder, parameter values in a sequence parameter set for a sequence of pictures;
1[b]	defining, in the encoder, parameter values in a picture parameter set for a picture; and
1[c]	defining, in the encoder, 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.
Claim 2	
2	The method according to claim 1, wherein the slice header is included with a reference to a picture parameter set.
Claim 3	
3	The method according to claim 1, wherein the picture parameter set is included with a reference to a sequence parameter set.
Claim 4	
4	The method according to claim 1, wherein the picture parameter value is selected from a group of information of a picture order count and a frame number.

Claim 5	
5	The method according to claim 1, wherein the sequence parameter set and picture parameter set are transmitted less often than once per each picture.
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	
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 10	
10[pre]	An encoder for encoding sequences of pictures into a bitstream, wherein parameters are defined in a parameter set and each picture comprises information of one or more slices, the encoder comprising:
10[a]	a processor configured for:
10[b]	defining parameter values in a sequence parameter set for a sequence of pictures;
10[c]	defining parameter values in a picture parameter set for a picture; and
10[d]	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.
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 12	
12[pre]	A system comprising an encoder for encoding sequences of pictures into a bitstream, a decoder for decoding sequences of pictures from the bitstream, wherein parameters are defined in a parameter set and each picture comprises information of one or more slices, the encoder comprising:
12[a]	a processor configured for:
12[b]	defining parameter values in a sequence parameter set for a sequence of pictures;
12[c]	defining parameter values in a picture parameter set for a picture; and

12[d]	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;
12[e]	and the decoder comprising:
12[f]	a processor configured for:
12[g]	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;
12[h]	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;
12[i]	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
12[j]	using the at least one second picture parameter value in decoding.
Claim 13	
13	The system according to claim 12, wherein a parameter set is stored in both the encoder and the decoder.
Claim 14	
14[pre]	A transmitting device comprising an encoder for encoding sequences of pictures into a bitstream, wherein parameters are defined in a parameter set and each picture comprises information of one or more slices, the encoder comprising:

14[a]	a processor configured for:
14[b]	defining parameter values in a sequence parameter set for a sequence of pictures;
14[c]	defining parameter values in a picture parameter set for a picture; and
14[d]	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.
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:
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 16	
16[pre]	A bitstream comprising:
16[a]	encoded pictures;
16[b]	sequence parameters in a sequence parameter set for a sequence of pictures;
16[c]	picture parameter values in a picture parameter set for a picture;
16[d]	information of a slice comprising a slice header; and
16[e]	at least one picture parameter value in the slice header, the picture parameter value remaining unchanged at least in all slice headers of one encoded picture.
Claim 17	
17	The bitstream according to claim 16, wherein the slice header is included with a reference to a picture parameter set.
Claim 18	
18	The bitstream according to claim 16, wherein the picture parameter set is included with a reference to a sequence parameter set.
Claim 19	
19	The bitstream according to claim 16, wherein the picture parameter value is selected from a group of information of a picture order count and a frame number.

Claim 20	
20[pre]	A computer readable storage medium embedded with a computer program comprising programming code for execution on a processor, the programming code for encoding sequences of pictures into 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:
20[a]	defining parameter values in a sequence parameter set for a sequence of pictures;
20[b]	defining parameter values in a picture parameter set for a picture; and
20[c]	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.
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
21[d]	using the at least one second picture parameter value in decoding.
Claim 22	
22[pre]	An encoder for encoding sequences of pictures into a bitstream, wherein parameters are defined in a parameter set and each picture comprises information of one or more slices, the encoder comprising:
22[a]	means for defining parameter values in a sequence parameter set for a sequence of pictures;
22[b]	means for defining parameter values in a picture parameter set for a picture; and
22[c]	means for 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.
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

Snap Inc. ("Snap" or "Petitioner") requests *inter partes* review ("IPR") of claims 1–23 ("Challenged Claims") of U.S. Patent 7,724,818.

II. GROUNDS FOR STANDING

The '818 patent is available for IPR and Petitioner is not barred or estopped from requesting IPR on the grounds herein. An IPR was previously filed and instituted for the '818 patent but was terminated due to settlement between the parties. *See* IPR2024-00572, paper no. 29. This petition is substantially similar to the previously instituted petition. *See* IPR2024-00572, paper no. 3. Snap had no involvement, and was not a real party in interest in the previous IPR.

III. IDENTIFICATION OF CHALLENGE

A. Prior Art[1]

Petitioner requests review and cancellation of the Challenged Claims under:

Prior Art
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)

[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.

Prior Art
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	1–23	Au
2	§103	1–23	VCEG-N52, Sriram

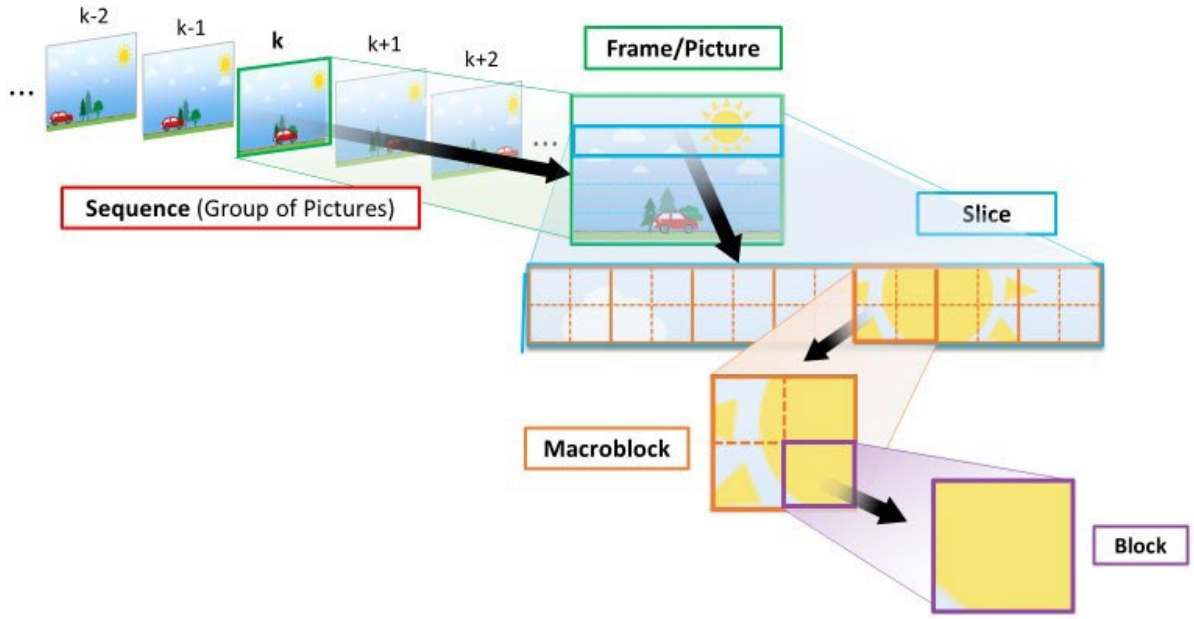
IV. OVERVIEW

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/decoders have been widely used since at least 1995. Saber, ¶45 (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, ¶46 (citing EX-1015, 10, 13).



The ’818 patent describes the “Parameter Set Concept,” which it acknowledges was described in VCEG-N52.[2] EX-1001, 2:48-3:28; EX-1012, 8; Saber, ¶47. According to the ’818 patent, VCEG-N52 included a parameter set with “all picture, GOP, and sequence level data” and a unique identifier. EX-1001, 2:53–57. Further, in VCEG-N52, “[e]ach slice header includes a reference to a parameter

[2] 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, Saber, n.2; *see also* EX-1013 (VCEG-N55 not related to parameters or parameter sets).

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 implements VCEG-N52's known parameter set concept. *Id*., 3:55–4:1; Saber, ¶48. As Dr. Saber explains, Figure 2 (below) depicts only conventional elements of an encoder/decoder system (encoder1/decoder2); Figure 4 (below) depicts only conventional decoder components (processor and memory); and VCEGN52's parameter set concept was implemented on such a conventional encoder/decoder system. Saber, ¶48.

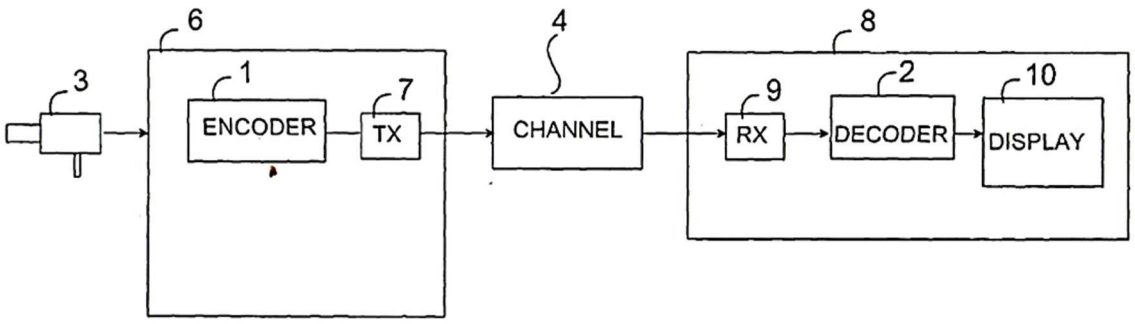

6
4
8
3
1
7
9
2
10
ENCODER
TX
CHANNEL
RX
DECODER
DISPLAY

Fig. 2

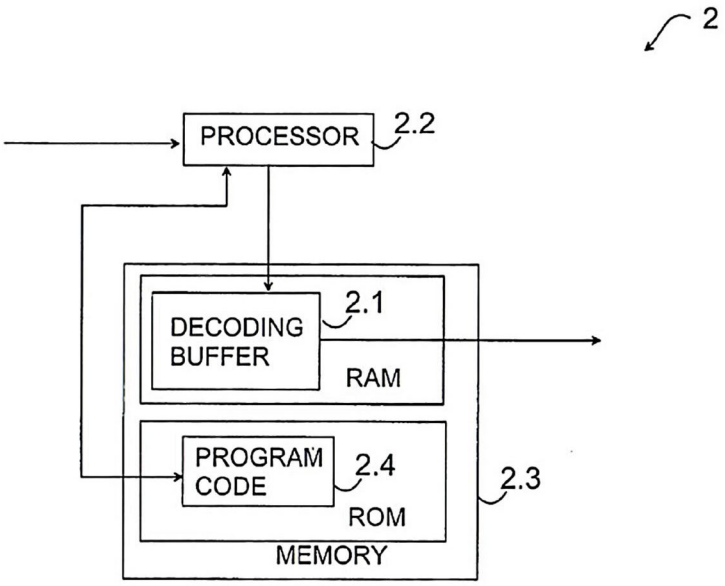

2
PROCESSOR
2.2
DECODING BUFFER
2.1
RAM
PROGRAM CODE
2.4
ROM
2.3
MEMORY

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,” “a picture parameter set,” and picture parameters in a “slice header.” *Id.*, Abstract, 4:41–48. Figure 1 of the ’818 patent, below, shows how the parameter sets (independent GOP parameter set 105 (red), picture parameter set 104 (green), and slice parameter set 103 (blue)), relate to a coded video sequence. Saber, ¶¶49–50.

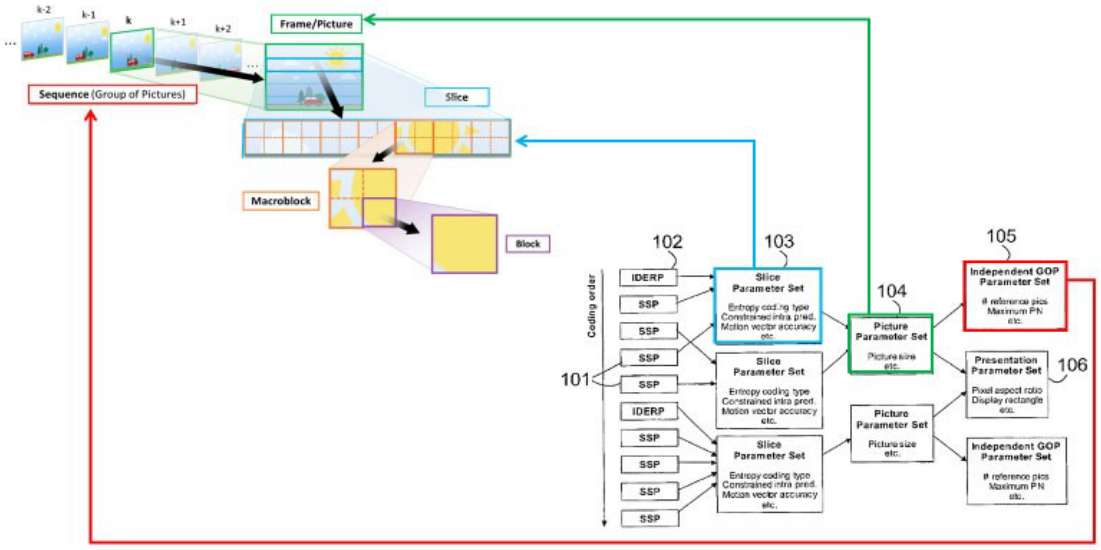


Fig. 1

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, ¶52. Although 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: sequence parameter set and forming [a] sequence parameter pertaining to a sequence using the parameter set; recognizing a picture parameter set and forming [a] first picture parameter pertaining to a picture using the parameter set; forming [a] second picture parameter using information of a slice header, the [a] second picture parameter remaining unchanged at least in all slice headers of one picture; and using the [a] second picture parameter in decoding." EX-1002, 31. Similar features were recited in other originally-filed independent claims. Saber, ¶53.

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, applicant argued Vishiram disclosed "no more than what is included in" VCEG-N52 and provided VCEG-N52 to the Examiner. *Id.*, 142; Saber, ¶54.

Over six years after the application was filed, applicant amended the pending independent claims corresponding to those challenged here to, *inter alia*, recite forming "picture parameter **values**" (not "picture parameters"), including that the "second picture parameter **value**…remain[s] unchanged at least in all slice headers of one picture." EX-1002, 403–04, 406–09. 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.

Afterwards Examiner allowed the claims without explanation. *Id.*, 417–20; Saber, ¶¶55–56.

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 (1) a Bachelor's degree in electrical engineering, computer engineering, computer science, or a comparable field of study such as physics; (2) approximately two to three years of practical experience with video encoding/decoding. Additional experience could substitute for the level of education and vice versa. Saber, ¶¶25–26.

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.[3] *Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999).

A. Means-Plus-Function (MPF)[4]

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 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).

---

[3] Petitioner reserves its right to address construction of any terms raised in the related actions.

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

Claims 22 and 23 recite limitations as "means for" performing a function.

For claim 22 Petitioner identifies "an encoder as described in the '818 patent specification in figure 3 and equivalents thereof" as the structure corresponding to the claimed functions of each "means for" limitation. Saber, ¶59. For claim 23 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.[5] *Id.*

VII. DETAILED EXPLANATION OF GROUNDS

A. Ground 1: 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, ¶¶61–63.

Au discloses "a hierarchy of syntax levels…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**

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

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[.]" *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. Additionally, 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, reproduced below.

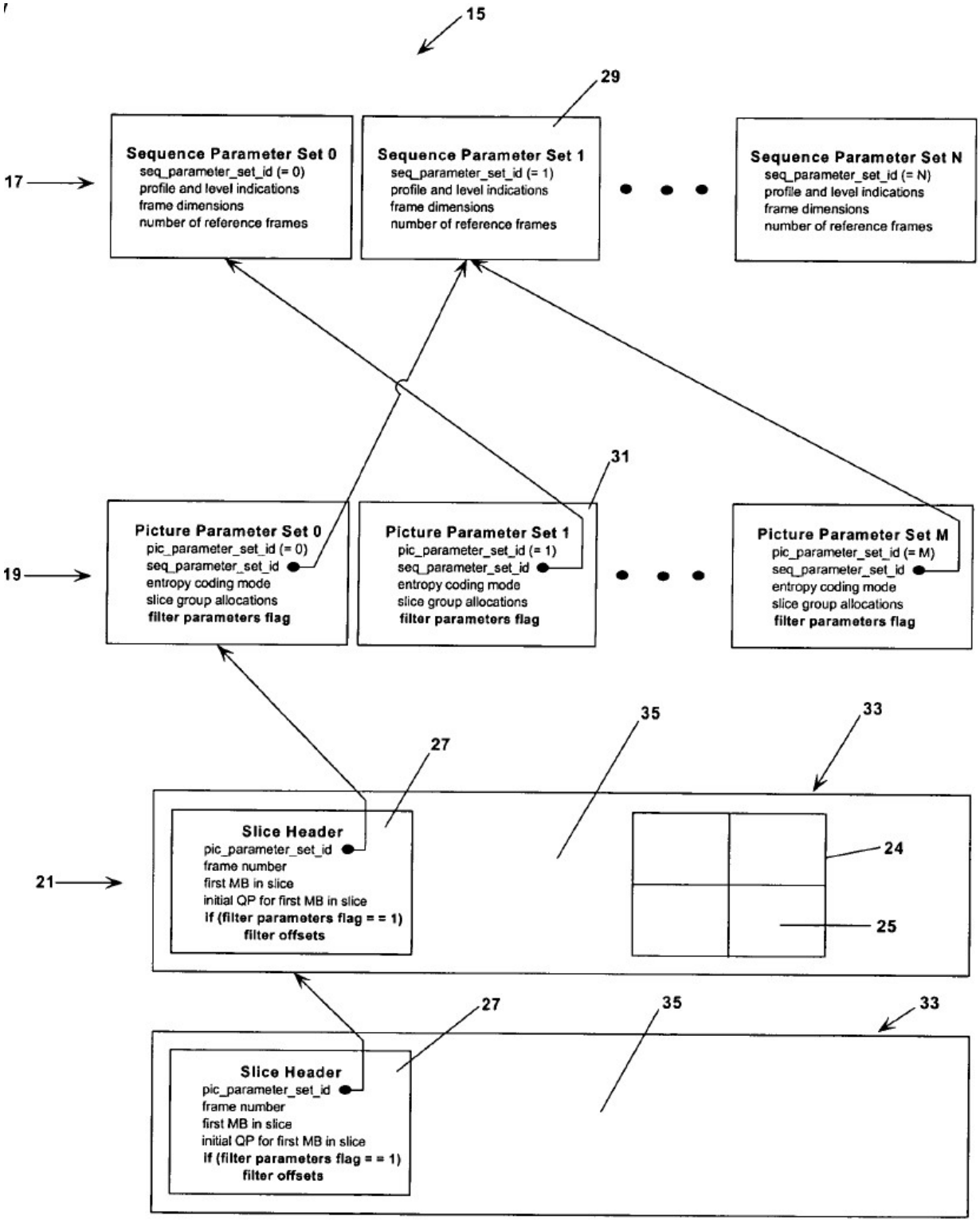

15
29
17
Sequence Parameter Set 0
seq_parameter_set_id (= 0)
profile and level indications
frame dimensions
number of reference frames
Sequence Parameter Set 1
seq_parameter_set_id (= 1)
profile and level indications
frame dimensions
number of reference frames
Sequence Parameter Set N
seq_parameter_set_id (= N)
profile and level indications
frame dimensions
number of reference frames
31
19
Picture Parameter Set 0
pic_parameter_set_id (= 0)
seq_parameter_set_id
entropy coding mode
slice group allocations
filter parameters flag
Picture Parameter Set 1
pic_parameter_set_id (= 1)
seq_parameter_set_id
entropy coding mode
slice group allocations
filter parameters flag
Picture Parameter Set M
pic_parameter_set_id (= M)
seq_parameter_set_id
entropy coding mode
slice group allocations
filter parameters flag
35
33
27
21
Slice Header
pic_parameter_set_id
frame number
first MB in slice
initial QP for first MB in slice
If (filter parameters flag = = 1)
filter offsets
24
25
27
35
33
Slice Header
pic_parameter_set_id
frame number
first MB in slice
initial QP for first MB in slice
If (filter parameters flag = = 1)
filter offsets

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, ¶¶61–63. Au also describes the '818 patent's purported invention: splitting the parameter sets into the "sequence parameter set" and the "picture parameter set," shown in Au's Figure 2 hierarchy above, and picture parameters (e.g., "frame_num") included in a "slice header." *Id*.

2. Claim 1

a. 1[pre]

If limiting, Au teaches 1[pre]. Saber, ¶¶64–67. Au discloses "a video conferencing system 10 has participants A and B that exchange video data 12 between monitors 13, formatted as a compressed bit stream 15 over a network 14." Au, 6:66–7:3, Fig. 1. Au discloses "processors 16 coordinate the display of successive frames 22 on the monitors 13, as the video exchange **of video data 12 in the compressed bit stream 15**."[6] Au, 7:13–16. Au also discloses an *encoder*. Au, Fig. 1, item 18, 7:22–24, Fig. 4. A POSITA would have understood such compression

[6] All emphasis/annotations added unless stated otherwise.

to be "*encoding a sequence of pictures into a bitstream*" because frames are synonymous with pictures in most digital video applications. Saber, ¶64.

Au discloses that "**'parameter sets'** allows efficient transmission of infrequently changing data at the sequence 17 and picture level 19 in the H.264 standard." Au, 8:8–11. Au discloses "parameter sets" that include "values **of parameters[.]**" Au, 8:9–19. A POSITA would have understood this to disclose "*parameters [which] are defined in a parameter set*." Saber, ¶65.

Au discloses a slice that "contains a picture data **35** representing a sub-set of the macroblocks **24** of the complete picture **22**." *Id.*, 8:36–38. Further, Au discloses "each **slice 33 of the frame 22 is encoded by the encoder 18** (see FIG. 1), independently from the other slices 33 in the frame 22." Au, 8:66–9:1, Fig. 2. Therefore, "*each picture comprises information of one or more slices*." Saber, ¶¶66–67.

b. 1[a]

Au teaches 1[a]. Saber, ¶¶68–70. Au discloses an *encoder*. Au, Fig. 1, item 18, 7:22–24, Fig. 4. Au discloses "[a] sequence parameter set 29 in the first level 17 includes values of parameters that will remain unchanged for an **entire video sequence**." Au, 8:11–15. A POSITA would have understood the video sequence to be a "*sequence of pictures*." Saber, ¶68; Au, 7:25–27. Au further provides "[e]xamples of parameters in a sequence parameter set 29 include frame dimensions

and the maximum number of reference frames." Au, 8:16–19, Fig. 2. Au also discloses "**encoder 18** [shown below] and decoder 20 could be to a storage device (not shown) for later usage" (Au, 7:22–24) and "[t]he choice of sequence parameter sets 29 and picture parameter sets 31 can be chosen by the encoder 18." Au, 9:13–17.

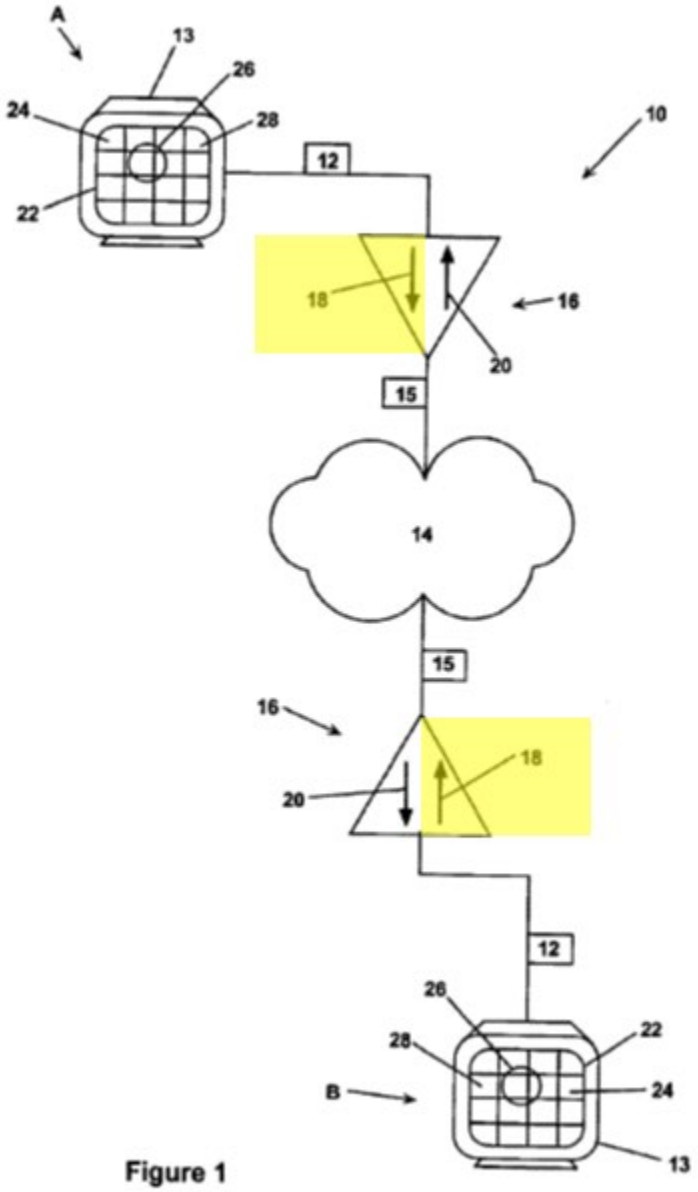


Figure 1

Au, Fig. 1. A POSITA would have understood the above as "*defining, in an encoder*" because the parameter values in the sequence parameter set are chosen by the encoder (18) or at minimum setup for operation of the encoder. Saber, ¶68.

Au explains: "[a] sequence parameter set 29 in the first level 17 **includes values of parameters** that will remain unchanged for an entire video sequence[.]" (*parameter values in a sequence parameter set for a sequence set*). Au, 8:9–19. Further, "the bitstream 15 is organizing into 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, 8:6–9, Fig. 2; Saber, ¶¶69–70.

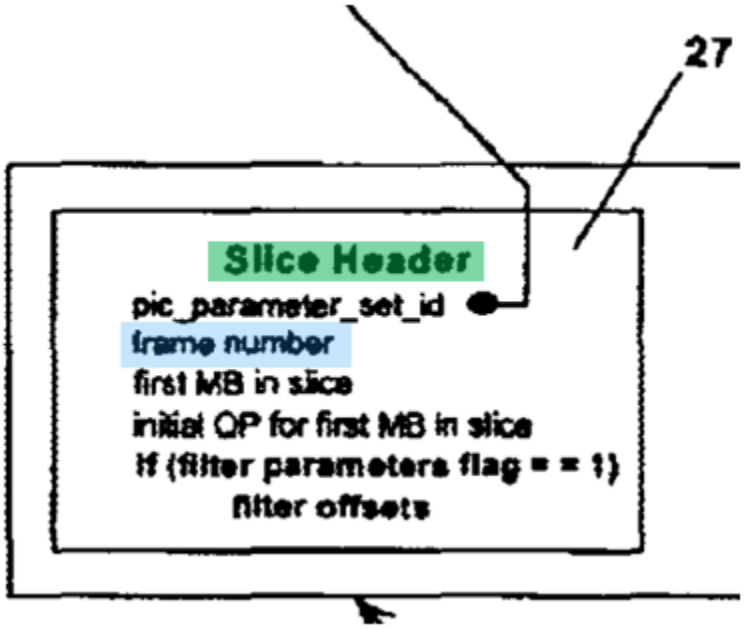
c. 1[b]

Au teaches 1[b]. Saber, ¶¶71–72. As discussed above, Au discloses "*defining, in the encoder*." Section VII.A.2.b; Saber, ¶¶68-70. Au discloses a "picture parameter set" that includes "**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**[,]" and "slice group allocations[.]" *Id.*, 8:23–26, Fig. 2. Further, Au 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" (*parameter values in a picture*

parameter set for a picture). Au, 9:8–13; Saber, ¶¶71–72.

d. 1[c]

Au teaches 1[c]. Saber, ¶¶73–75. Au discloses an encoder. Au, Fig. 1, item 18, 7:22–24, Fig. 4. As discussed above, Au discloses "*defining, in the encoder*." Section VII.A.2.b; Saber, ¶¶68-70. Au teaches "*pictures*" and "*slice header*." Specifically, Au discloses a "picture" is a "frame or field." Au, 8:22. Au discloses "[t]he coded slice 33 includes the **slice header** 27 and the slice data 35." Au, 8:39–40. Au also teaches information of a slice header in the form of "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. Figure 2, partially reproduced below, highlights frame number (blue) and slice header (green). Saber, ¶73. A POSITA would have understood that a frame number is a picture parameter value as numbers are "*values*," and therefore meets the limitation of "*at least one picture parameter value in a slice header.*" *Id.*



As to "*the picture parameter value remaining unchanged at least in all slice headers of one picture*," Au discloses a slice "contains a 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. When frame_num is used, each slice of the picture **would have the same frame number** in order to properly reconstruct the frame. Saber, ¶74. That frame number (the "*picture parameter value*") would thus "*remain[] unchanged at least in all slice headers of one picture*." frame_num would function this way to ensure that all the slices that belong to a given frame are indexed accordingly in order to spatially position them correctly within the frame for proper reconstruction. Saber, ¶¶74–75.

3. Claim 2

Au teaches claim 2 for the same reasons as claim 7. Section VII.A.8; Saber, ¶76.

4. Claim 3

Au teaches claim 3 for the same reasons as claim 8. Section VII.A.9; Saber, ¶77.

5. Claim 4

Au teaches claim 4 for the same reasons as claim 9. Section VII.A.10; Saber, ¶78.

6. Claim 5

Au teaches claim 5. Saber, ¶¶79–81. Section VII.A.2 explains that Au teaches the "method according to claim 1." Discussed in Section VII.A.2.b (1[a]), Au discloses "'parameter sets' allow efficient transmission of infrequently changing data at the sequence 17 and picture level 19 in the H.264 standard." Au, 8:8–11. Au discloses "**[a] sequence parameter set 29** in the first level 17 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, 8:11–19.

Further, Au discloses "**[a] picture parameter set 31** in the second level 21 includes values of parameters that will remain unchanged within a coded representation of a picture (frame or field) 22." *Id.*, 8:20–22. A POSITA would have

understood that because the sequence parameter set/picture parameter set remain unchanged within a coded representation of a picture and may be the same set used in follow on picture(s), they are "*transmitted less often than once per each picture*." Saber, ¶¶79–81.

7. Claim 6

a. 6[pre]

If limiting, Au teaches 6[pre]. Saber, ¶¶82–85. Au discloses a "method…for block-based motion-compensated decoding." Au, Abstract. Au 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, ¶82.

Au 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, ¶83. Further, 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 of one or more slices*." Saber, ¶¶84–85.

b. 6[a]

Au teaches 6[a]. Saber, ¶¶86–88. 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, ¶86. Au discloses a "sequence parameter set" including "values of parameters that will remain unchanged for an entire video sequence, or from one [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," and "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 that (1) "*recognizing, in a decoder*" encompasses a decoder parsing the incoming bitstream and identifying the recited "*sequence parameter set*;" and (2) "*forming*" 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*;" and therefore, 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. Saber, ¶¶87–88.

c. 6[b]

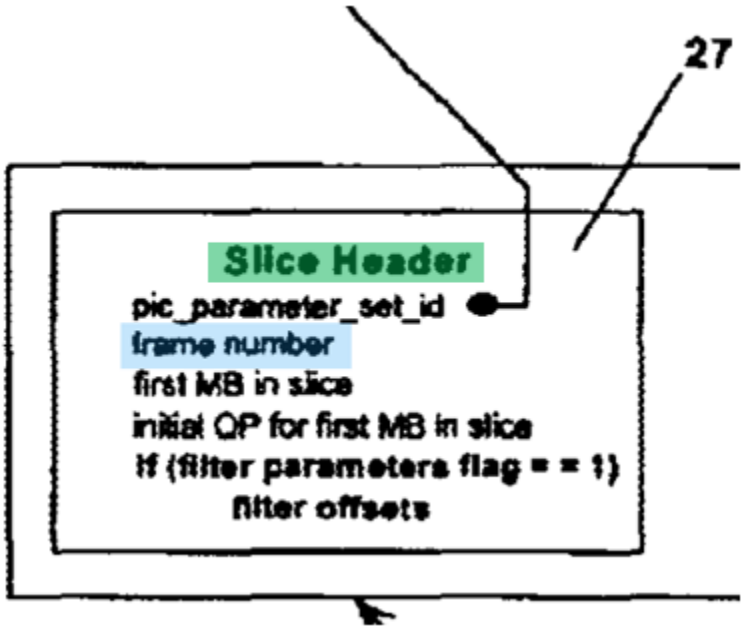
Au teaches 6[b]. Saber, ¶¶89–91. Au discloses a "picture parameter set," including "values of parameters that will remain unchanged within a coded representation of a picture (frame or field) **22**." Au, 8:20–22. Au discloses examples of parameters in a picture parameter set, including "the entropy coding mode and a flag that specifies whether deblocking filter parameters will be transmitted in the slice headers **27** of the picture **22**[,]" and "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." *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, ¶89.

A POSITA would have understood that (1) "*recognizing, in the decoder*" encompasses a decoder parsing the incoming bitstream and identifying the recited "*picture parameter set*;" and (2) "*forming*" 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*;" and therefore, 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. *Id*., ¶¶90–91.

d. 6[c]

Au teaches 6[c]. *Id*., ¶¶92–95. Au teaches "pictures" (Au, 8:22) and "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" (blue).



A POSITA would have understood 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, ¶93.

A POSITA would have understood Au teaches that (1) "*forming, in the decoder*" encompasses a decoder creating in local memory the value associated with the data element, such as, "*at least one second picture parameter value using information of a slice header*;" and (2) 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.*, ¶94.

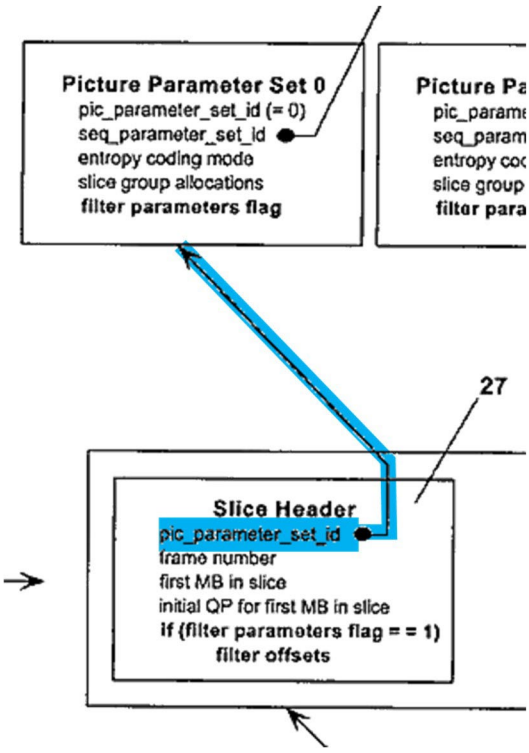
Au discloses a slice "contains picture data 35 representing a sub-set of the macroblocks 24 of the complete picture 22." Au, 8:36–38. Au discloses a picture is a "frame or field." *Id.*, 8:22; *see also id.*, 7:25–26. Therefore, Au discloses a picture that is a frame, the data of which is included in slices. Saber, ¶95. When 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 "*second picture parameter value*," would thus "*remain[] unchanged at least in all slices of one picture*." *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*.

e. 6[d]

Au teaches 6[d]. *Id*., ¶96. Au discloses 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, ¶96. 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, as the "*second picture parameter value*," in decoding. *Id*.

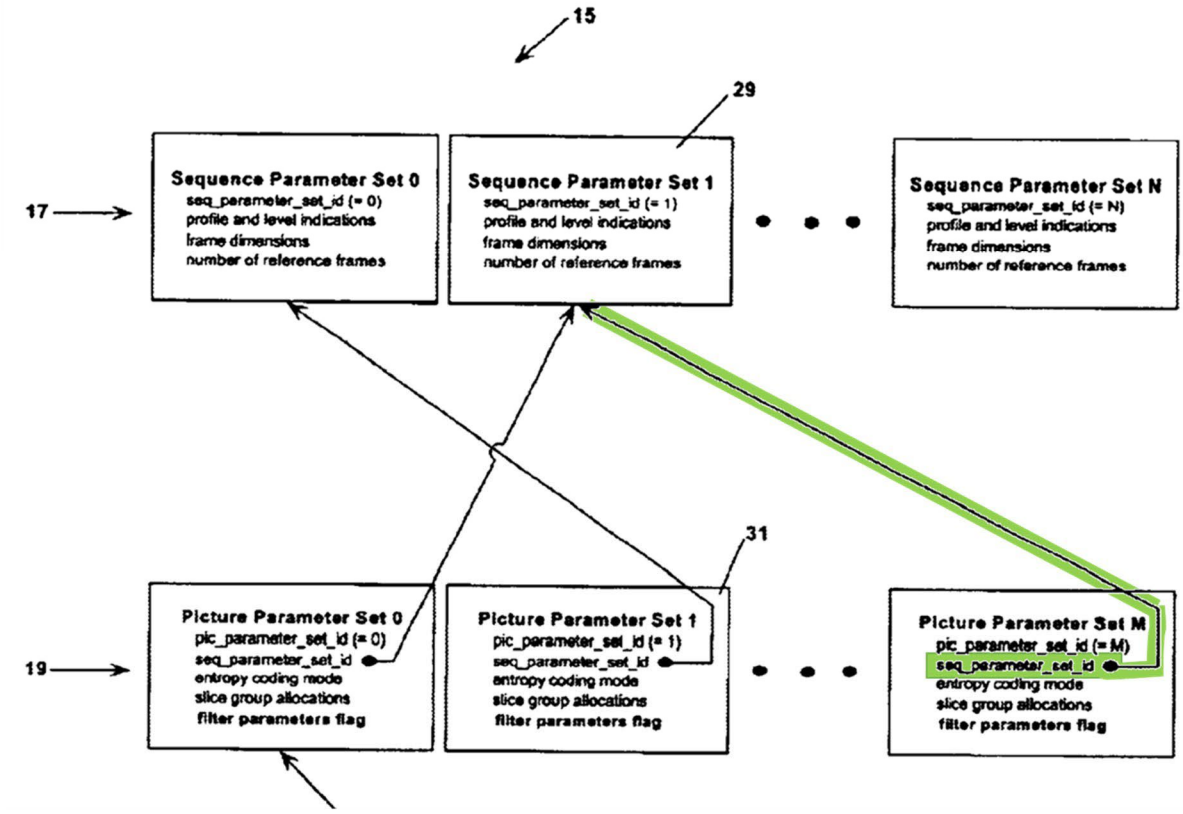
8. Claim 7

Au teaches claim 7. Saber, ¶¶97–98. Section VII.A.7 explains Au teaches the claim 6. As discussed for 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. Figure 2 (relevant section reproduced below) shows the slice header includes "pic_parameter_set_id" with an arrow pointing to a picture parameter set. Saber, ¶¶97–98.



9. Claim 8

Au teaches claim 8. Saber, ¶¶99–100. Section VII.A.7 explains Au teaches the "method according to claim 6." As discussed in Section VII.A.7.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. Figure 2 (portion below) shows each picture parameter set includes "seq_parameter_set_id" with an arrow pointing to a sequence parameter set. Saber, ¶¶99–100.



10. Claim 9

Au teaches claim 9. Saber, ¶101. Section VII.A.7 explains Au teaches the claim 6. As discussed for 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*.

11. Claim 10

a. 10[pre]

If limiting, Au teaches 10[pre] for the same reasons as 1[pre]. Section

VII.A.2.a; Saber, ¶102.

b. 10[a]

Au teaches 10[a] for the same reasons as 11[a]. Section VII.A.12.a; Saber, ¶103.

c. 10[b]

Au teaches 10[b] for the same reasons as 1[a]. Section VII.A.2.b; Saber, ¶104.

d. 10[c]

Au teaches 10[c] for the same reasons as 1[b]. Section VII.A.2.c; Saber, ¶105.

e. 10[d]

Au teaches 10[d] for the same reasons as 1[c]. Section VII.A.2.d; Saber, ¶106.

12. Claim 11

a. 11[pre]

If limiting, Au teaches 11[pre]. Saber, ¶107. As discussed in Section VII.A.7.a (6[pre]), Au teaches "*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 discloses this decoding is done by a decoder. Au, 6:54–55, 10:41–46 ("The decoder **20** [] of the receiving participant A,B processes the received bit stream 15 and then reconstructs the coded frame[.]"), Fig. 3; Saber, ¶107.

b. 11[a]

Au teaches 11[a]. Saber, ¶108. Au discloses that the processors coordinate the display of successive frames on monitors as video data is communicated between participants A, B during a video conference. Au, 7:11–14. This is illustrated in Figure 1 below, processors (16) in yellow.

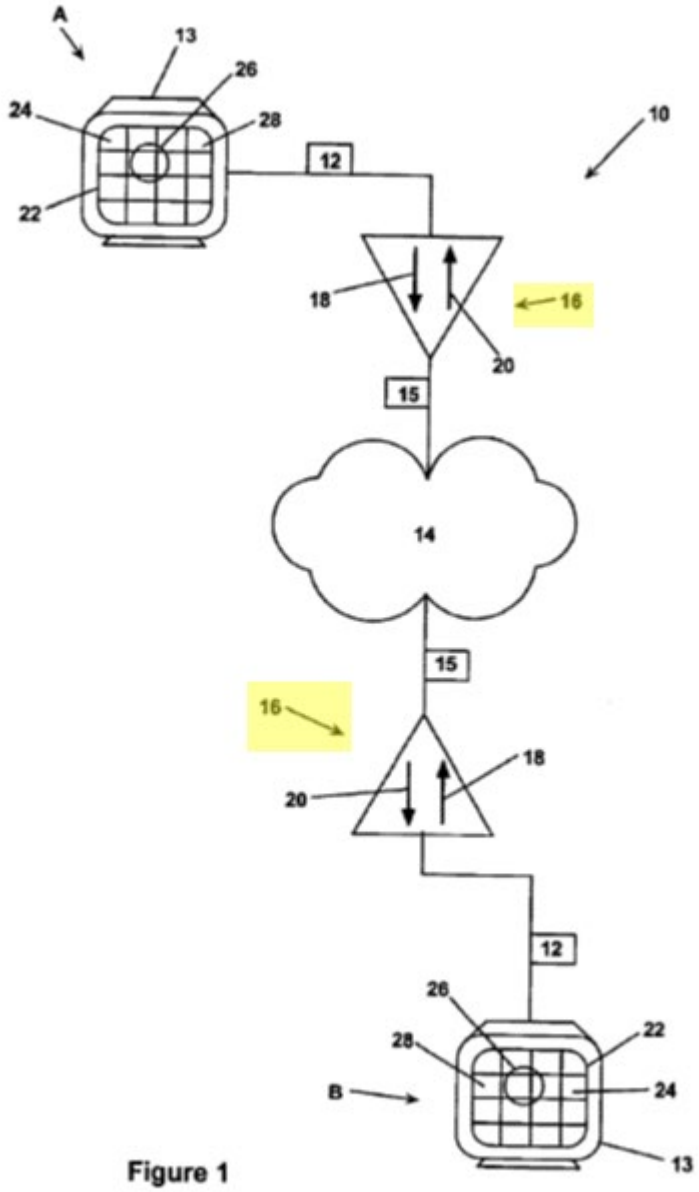


Figure 1

Au's decoder 20 comprises a processor 16. Saber, ¶108. 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.A.7.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.*, ¶109. Au discloses a sequence parameter set "includes values of parameters that will remain unchanged for an entire video sequence, or from one [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*." Saber, ¶109. 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*," because that parameter is used for displaying pictures in the correct dimensions. *Id.*; Au, 8:16–19, Fig. 2.

d. 11[c]

Au teaches 11[c] for the same reasons as 6[b]. Section VII.A.7.c; Saber, ¶110.

e. 11[d]

Au teaches 11[d] for the same reasons as 6[c]. Section VII.A.7.d; Saber, ¶111.

f. 11[e]

Au teaches 11[e] for the same reasons as 6[d]. Section VII.A.7.e; Saber, ¶112.

13. Claim 12

a. 12[pre]

If limiting, Au teaches 12[pre]. Saber, ¶¶113–16.

As discussed in Section VII.A.2.a (1[pre]), Au teaches "*an encoder for encoding sequences of pictures into a bitstream*," and "*wherein parameters are defined in a parameter set and each picture comprises information of one or more slices*." Saber, ¶114.

As discussed in Section VII.A.12.a (11[pre]), Au teaches "*a decoder for decoding sequences of pictures from the bitstream*." Saber, ¶115.

Au also discloses a system comprising an "encoder 18" and a "decoder 20." Au, 6:66–7:24, Fig. 1. Figure 1 (shown below, encoders in yellow) is described by Au as "a diagram of an encoding/decoding system." Au, 6:52; Saber, ¶116.

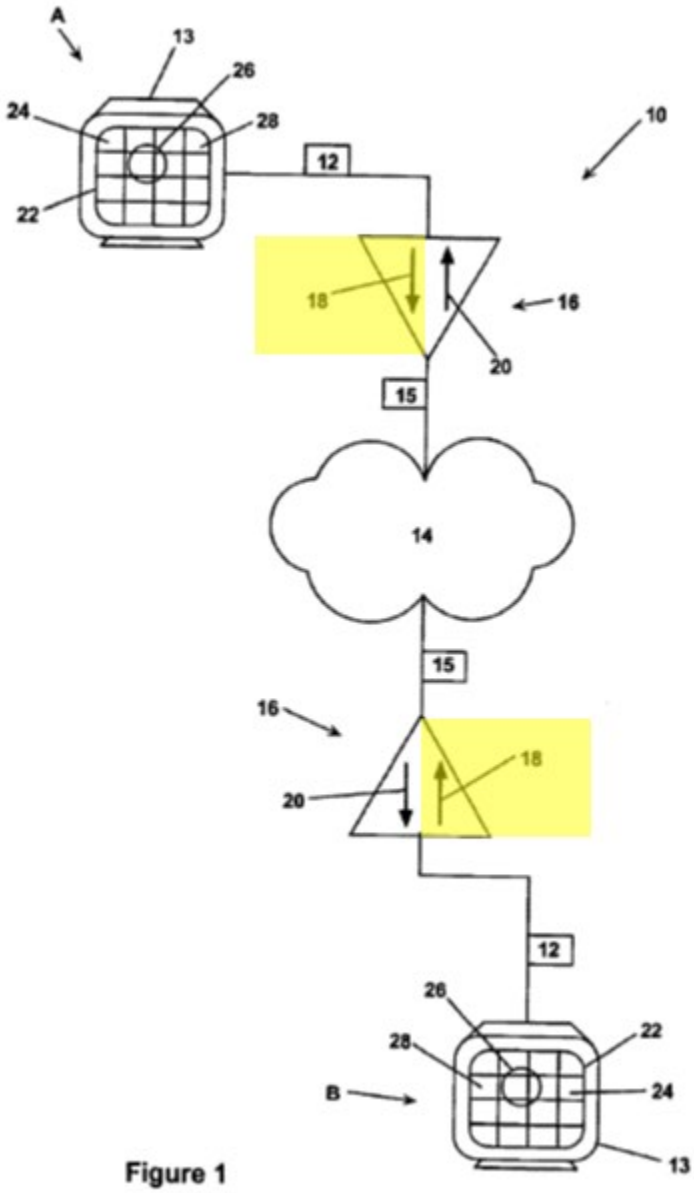


Figure 1

b. 12[a]

Au teaches 12[a] for the same reasons as 10[a] and 11[a]. Sections VII.A.11.b, VII.A.12.b; Saber, ¶117.

c. 12[b]

Au teaches 12[b] for the same reasons as 1[a]. Section VII.A.2.b; Saber, ¶118.

d. 12[c]

Au teaches 12[c] for the same reasons as 1[b]. Section VII.A.2.c; Saber, ¶119.

e. 12[d]

Au teaches 12[d] for the same reasons as 1[c]. Section VII.A.2.d; Saber, ¶120.

f. 12[e]

Au teaches 12[e] for the same reasons as 11[a]. Section VII.A.12.b; Saber, ¶121.

g. 12[f]

Au teaches 12[f] for the same reasons as 10[a] and 11[a]. Sections VII.A.11.b, VII.A.12.b; Saber, ¶122.

h. 12[g]

Au teaches 12[g] for the same reasons as discussed in 6[a] and 11[b]. Sections VII.A.7.b, VII.A.12.c; Saber, ¶123.

i. 12[h]

Au teaches 12[h] for the same reasons as discussed in 6[b] and 15[c]. Sections VII.A.7.c, VII.A.16.d; Saber, ¶124.

j. 12[i]

Au teaches 12[i] for the same reasons as discussed in 6[c] and 11[d]. Sections VII.A.7.d, VII.A.12.e; Saber, ¶125.

k. 12[j]

Au teaches 12[j] for the same reasons as discussed in 6[d] and 11[e]. Sections VII.A.7.e, VII.A.12.f; Saber, ¶126.

14. Claim 13

Au teaches claim 13. Saber, ¶¶127–28. Section VII.A.13 explains Au teaches the "method according to claim 12." Au discloses that "[f]urther compression of the video information is realized by entropy coding the resulting quantized transform coefficients before transmission **or storage** of the encoded bit stream." Au, 1:41–44, 7:22–24. Au discloses that parameter sets allow "for efficient transmission of infrequently changing data at the sequence 17 and picture level 19" and that "[a] sequence parameter set 29 in the first level 17 includes values of parameters that will remain unchanged for an entire video sequence[.]" Au, 8:9–19.

A POSITA would have understood that given the disclosed storage capabilities and because the sequence parameter set includes values of parameters that remain unchanged throughout an entire video sequence, these would need to be stored in **both** the encoder and decoder. Saber, ¶128.

15. Claim 14

a. 14[pre]

If limiting, Au teaches 14[pre] for the same reasons as 1[pre], 10[pre], and 12[pre]. Sections VII.A.2.a, VII.A.11.a, VII.A.13.a; Saber, ¶¶129–30.

Au discloses a system comprising an "encoder 18" and a "decoder 20." Au, 6:66–7:24, Fig. 1, 7:12–14. A POSITA would have understood that communication between the participants over the network 14 (*see* Fig. 1) would require a transmitting device. Au goes on to state that "[t]he encoder 18 [] emulates the behaviour of the decoder 20 for coded blocks 22 to make sure the encoder 18 of the **transmitting** participant A,B and the decoder 20 of the receiving participant A,B work from the same reference frames 22b." Au, 10:33–37; Saber, ¶130.

b. 14[a]

Au teaches 14[a] for the same reasons as 11[a]. Section VII.A.12.b; Saber, ¶131.

c. 14[b]

Au teaches 14[b] for the same reasons as 1[a]. Section VII.A.2.b; Saber, ¶132.

d. 14[c]

Au teaches 14[c] for the same reasons as 1[b]. Section VII.A.2.c; Saber, ¶133.

e. 14[d]

Au teaches 14[d] for the same reasons as 1[c]. Section VII.A.2.d; Saber, ¶134.

16. Claim 15

a. 15[pre]

If limiting, Au teaches 15[pre]. Saber, ¶¶135–36. Beyond 11[pre] (Section VII.A.12.a), 15[pre] adds only a "*receiving device*" comprising 11[pre]'s decoder.

Au 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 decoding in Au is performed in such devices because the received data is what is decoded. Saber, ¶135.

b. 15[a]

Au teaches 15[a] for the same reasons as 11[a]. Section VII.A.12.b; Saber, ¶137.

c. 15[b]

Au teaches 15[b] for the same reasons as 11[b]. Section VII.A.12.c; Saber, ¶138.

d. 15[c]

Au teaches 15[c] for the same reasons as 11[c]. Section VII.A.12.d; Saber, ¶139.

e. 15[d]

Au teaches 15[d] for the same reasons as 11[d]. Section VII.A.12.e; Saber, ¶140.

f. 15[e]

Au teaches 15[e] for the same reasons as 11[e]. Section VII.A.12.f; Saber, ¶141.

17. Claim 16

a. 16[pre]

If limiting, Au teaches 16[pre] for the same reasons as 1[pre]. Section VII.A.2.a; Saber, ¶142. Au states, referring to Figure 2, that "the bitstream 15 is organizing into 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, 8:6–9.

b. 16[a]

Au teaches 16[a] for the same reasons as 1[pre]. Section VII.A.2.a; Saber, ¶143.

c. 16[b]

Au teaches 16[b] for the same reasons as 1[a]. Section VII.A.2.b; Saber, ¶144.

d. 16[c]

Au teaches 16[c] for the same reasons as 1[b]. Section VII.A.2.c; Saber, ¶145.

e. 16[d]

Au teaches 16[d] for the same reasons as 1[c]. Section VII.A.2.d; Saber, ¶146.

f. 16[e]

Au teaches 16[e] for the same reasons as 1[c]. Section VII.A.2.d; Saber,

¶147.

18. Claim 17

Au teaches claim 17 for the same reasons as claim 7. Section VII.A.8; Saber, ¶148.

19. Claim 18

Au teaches claim 18 for the same reasons as claim 8. Section VII.A.9; Saber, ¶149.

20. Claim 19

Au teaches claim 19 for the same reasons as claim 9. Section VII.A.10; Saber, ¶150.

21. Claim 20

a. 20[pre]

If limiting, Au teaches 20[pre]. Saber, ¶¶151–54.

As discussed for 21[pre], Au teaches "*[a] computer readable storage medium embedded with a computer program comprising programming code for execution on a processor, the programming code for*[.]" Saber, ¶151.

As discussed for 1[pre], Au teaches "*encoding sequences of pictures into a bitstream, wherein parameters are defined in a parameter set and each picture comprises information of one or more slices*[.]" Saber, ¶152.

As discussed for 10[pre], Au's encoding is done by an encoder. Au, 6:56–57,

Fig. 4, 9:18–21. Au discloses that the processors coordinate the display of successive frames on monitors as video data is communicated between participants A,B during a video conference. *Id.*, 7:11–14. Au discloses the "bitstream is generated by the encoder." *Id.*, 14:1. Au discloses "[t]he encoder 18 (see FIG. 1) emulates the behaviour of the decoder 20 for coded blocks 22 to make sure the encoder 18 of the transmitting participant A,B and the decoder 20 of the receiving participant A,B work from the same reference frames 22b." Au, 10–33:37. A POSITA would have further understood that such an encoder using a processor would have been implemented with programming code in a computer program stored on a computer readable storage medium. Saber, ¶154.

b. 20[a]

Au teaches 20[a] for the same reasons as 1[a]. Section VII.A.2.b; Saber, ¶155.

c. 20[b]

Au teaches 20[b] for the same reasons as 1[b]. Section VII.A.2.c; Saber, ¶156.

d. 20[c]

Au teaches 20[c] for the same reasons as 1[c]. Section VII.A.2.d; Saber, ¶157.

22. Claim 21

a. 21[pre]

If limiting, Au teaches 21[pre]. Saber, ¶¶158–59. Beyond 6[pre] (Section VII.A.7.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 for 11[pre], Au's decoding is done by a decoder. Au, 6:54–55, 10:41–46, Fig. 3; Saber, ¶159. Au discloses that the processors coordinate the display of successive frames on monitors as video data is communicated between participants A,B during a video conference. Au, 7:11–14. As discussed in Section VII.A.12.b (11[a]), a POSITA would have understood Au discloses a processor for decoding video and 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, ¶159.

b. 21[a]

Au teaches 21[a] for the same reasons as 11[b]. Section VII.A.12.c; Saber, ¶160.

c. 21[b]

Au teaches 21[b] for the same reasons as 6[b]. Section VII.A.7.c; Saber, ¶161.

d. 21[c]

Au teaches 21[c] for the same reasons as 6[c]. Section VII.A.7.d; Saber, ¶162.

e. 21[d]

Au teaches 21[d] for the same reasons as 6[d]. Section VII.A.7.e; Saber, ¶163.

23. Claim 22

a. 22[pre]

If limiting, Au teaches 22[pre] for the same reasons as 1[pre]. Section VII.A.2.a; Saber, ¶164.

b. 22[a]

Au teaches 22[a] regardless of whether it is construed as MPF. Saber, ¶165. Au teaches 22[a] for the same reasons as 1[a]. Section VII.A.2.b; Saber, ¶165. Figure 3 of the ’818 patent, shown below, discloses a structure that includes a processor, encoding buffer, and program code (the encoding buffer and program code both stored in memory, *see also, e.g.*, Au, 11:2–9).

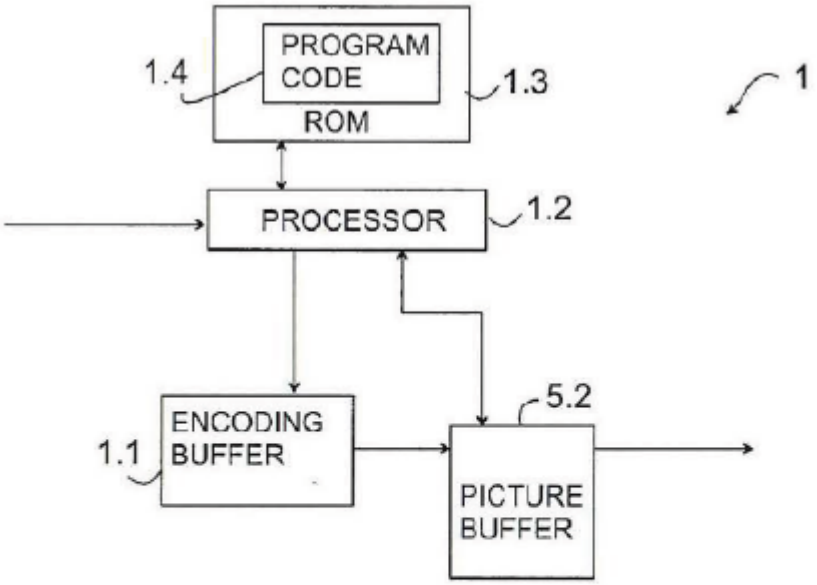


Fig. 3

As discussed in Section VII.A.2.b (1[a]), Au discloses "*defining, in an encoder, parameter values in a sequence parameter set for a sequence of pictures*." Au further discloses "the encoder…has an input frame **memory 30**…(see FIG. 1)." Au, 9:18–21. Moreover, Au discloses "processors 16 coordinate the display of success frames." Au, 7:10–16. A POSITA would have understood Au disclosed an encoder with structure the same as, or equivalent to, that of Figure 3 of the '818 patent, and that decoder performs the recited function in 23[a]. Saber, ¶166.

c. 22[b]

Au teaches 22[b] regardless of whether it is construed as MPF. Saber, ¶167. Au teaches 22[b] for the same reasons as 1[b]. Section VII.A.2.c; Saber, ¶167. If this term is construed as MPF, Au's decoder teaches the structure discussed in Section VII.A ("an encoder as described in the '818 patent specification in figure 3 and equivalents thereof") for the reasons discussed in Section VII.A.23.b (22[a]) above, and that encoder performs the recited function in 22[b]. Saber, ¶167.

d. 22[c]

Au teaches 22[c] regardless of whether it is construed as MPF. Saber, ¶168. Au teaches 22[c] for the same reasons as 1[c]. Section VII.A.2.d; Saber, ¶168. If this term is construed as MPF, Au's decoder teaches the structure discussed in Section VII.A ("an encoder as described in the '818 patent specification in figure 3 and equivalents thereof") for the reasons discussed in Section VII.A.23.b (22[a]) above, and that encoder performs the recited function in 22[c]. Saber, ¶168.

24. Claim 23

a. 23[pre]

If limiting, Au teaches 23[pre] for the same reasons as 11[pre]. Section VII.A.12.a; Saber, ¶169.

b. 23[a]

Au teaches 23[a] regardless of whether it is construed as MPF. Saber, ¶¶170–73. Au teaches the function as discussed in Section VII.A.12.c (11[b]). *Id.*, ¶171. If this term is construed as MPF, Au teaches the structure discussed in Section VI.A ("a decoder as described in the '818 patent specification in figure 4 and equivalents thereof"). *Id*., ¶172. As shown below, the '818 patent's Figure 4 discloses a structure including a processor, decoding buffer, and program code (decoding buffer and program code both stored in memory).

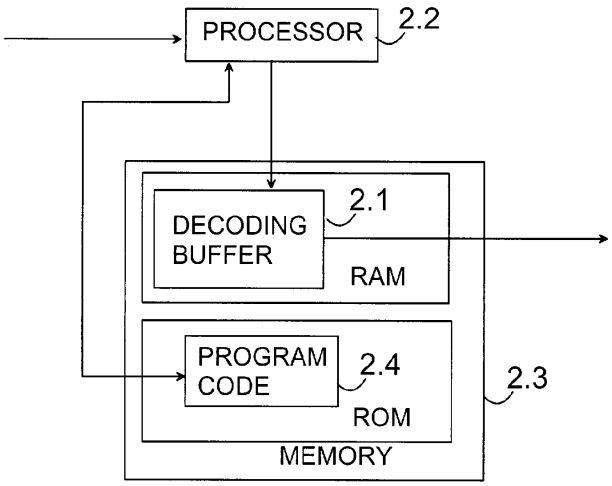


Fig. 4

As discussed in Section VII.A.12.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 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.A.12.b (11[a]) explains, Au discloses the decoder includes a processor. As discussed in Section VII.A.22.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 discloses the decoder buffer is stored using memory. *Id.*, 12:20–24 (discussing how decoder buffer affects "memory requirements"). A POSITA would have understood that 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, ¶173.

c. 23[b]

Au teaches 23[b] regardless of whether it is construed as MPF. *Id.*, ¶¶174–76. Au teaches the function as discussed in Section VII.A.7.c (6[b]). *Id.*, ¶175. If this term is construed as MPF, Au's decoder teaches 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*., ¶176.

d. 23[c]

Au teaches 23[c] regardless of whether it is construed as MPF. *Id.*, ¶¶177–79. Au teaches the function as discussed in Section VII.A.7.d (6[c]). *Id.*, ¶178. If this term is construed as MPF, Au's decoder teaches 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*., ¶179.

e. 23[d]

Au teaches 23[d] regardless of whether it is construed as MPF. *Id.*, ¶¶180–82. Au teaches the function as discussed in Section VII.A.7.e (6[d]). *Id.*, ¶181 If this term is construed as MPF, Au's decoder teaches 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*., ¶182.

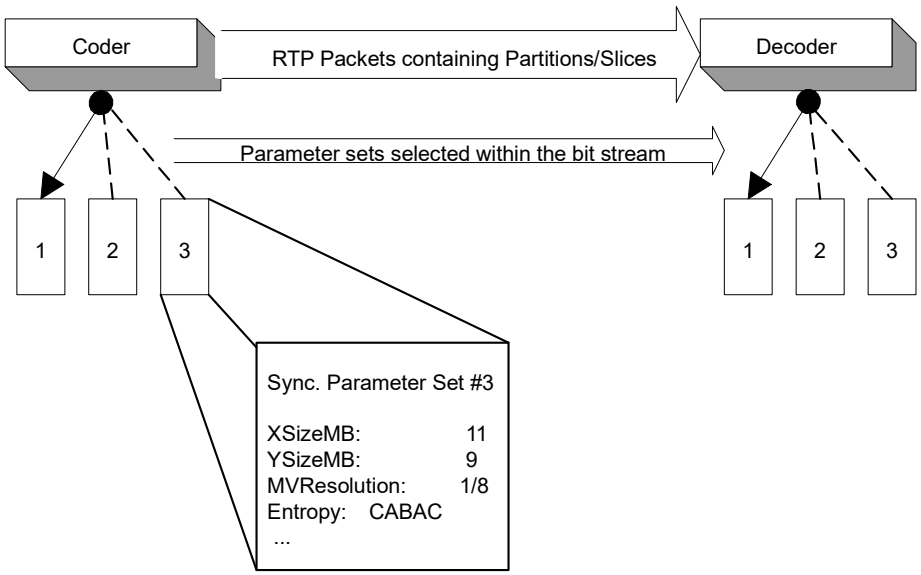
B. Ground 2: Obviousness over VCEG-N52 in view of Sriram

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. Hall-Ellis, ¶¶17–22. Further, VCEG-N52 itself indicates it was "Generated: 18 September '01," for a September 2001 meeting in Santa Barbara, California. *Id.*, ¶19. Therefore, VCEG-N52 was publicly available in September 2001 and is prior art under §102(b). *Id.*, ¶22; Saber, ¶184.

VCEG-N52 describes a "concept for the transport of H.26L over IP and H.324 networks" using parameter sets. VCEG-N52, 1; Saber, ¶185. 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, ¶187.

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, ¶188.

2. Sriram

Sriram discloses a video decoding system with an explicit "design goal" of "efficiency." Sriram, 1:43; Saber, ¶189. 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 that "[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.*, 13:29. 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, ¶190.

Sriram's slice data structure also includes a pointer to a picture frame data structure ("PicFrame *PicFramePtr"). Sriram, 15:57–65; Saber, ¶191. 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, ¶191. Further, because Sriram's "primary data structures" have pointers to "secondary data structures," a POSITA would have understood (or found it obvious) that 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, ¶192.

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, ¶¶193–98.

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, ¶193; (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, ¶193.

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, ¶194. 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, ¶194. 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, ¶194. 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, ¶¶193–94.

Extending VCEG-N52's parameter set concept with Sriram's multiple parameters would involve only routine engineering tasks that would yield predictable results. Saber, ¶¶195–96. 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*., ¶197. 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, ¶197. 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, ¶¶197–98.

4. Claim 1

a. 1[pre]

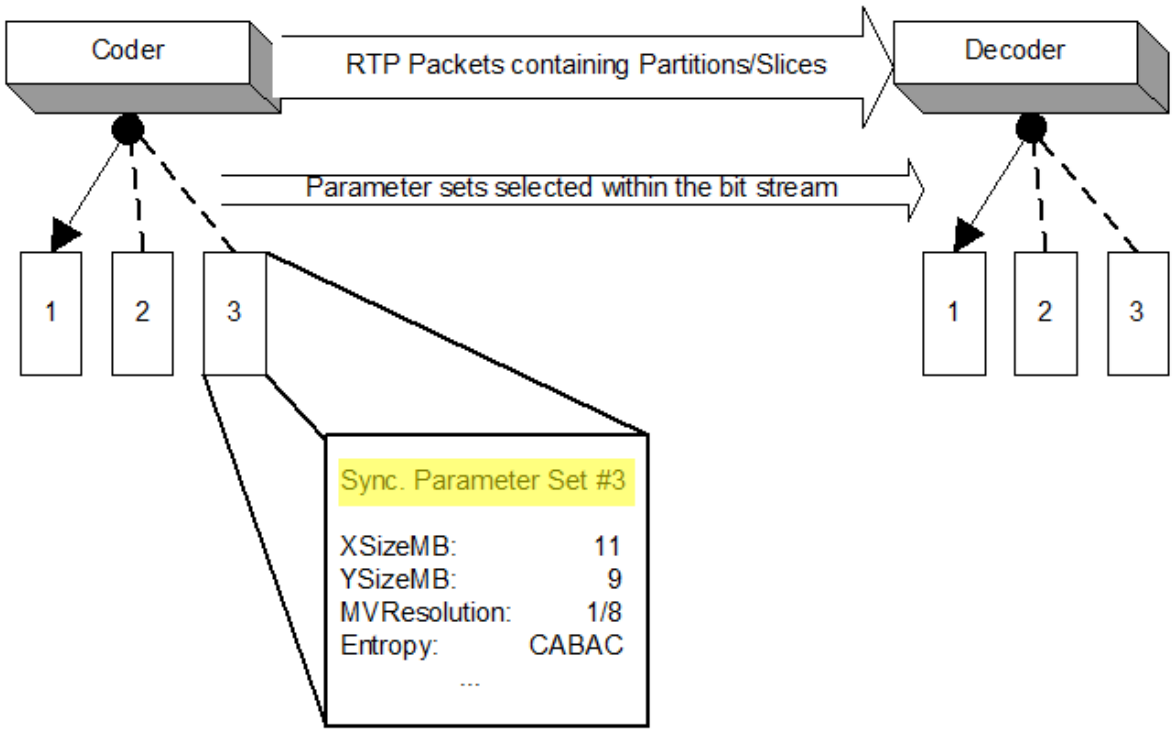
If limiting, combining VCEG-N52 and Sriram teaches 1[pre]. Saber, ¶¶199–202. VCEG-N52 and Sriram each disclose a video encoder. *See, e.g.*, VCEG-N52, 3, Fig. 1; Sriram, 4:21–27, Fig. 1. A POSITA would have understood the combination

would include an encoder and a method for encoding. Saber, ¶199.

VCEG-N52/Sriram disclose encoding pictures comprised of slices into a bitstream. Saber, ¶200; VCEG-N52, 3 ("the coder also generates side information"), 5, 6, 12 (describing the bitstream), 8 (describing that the encoder and decoder maintain a synchronized set of parameter sets, which contain all information associated with a picture and/or higher syntax layers), 1 ("RTP [Real-time Transit Procotol] stream" carried slices, and pictures are divided into slices); Sriram, 6:1–5 (describing a "hierarchy of data structures, depending on the level in which the parameters are applicable, is created in accordance with the invention," with "slice" coming just under "picture"). A POSITA would have understood that the "coder" referred to in VCEG-N52 is an encoder and the "encoder" referenced in Sriram provides the same functionality of "*encoding a sequence of pictures into a bitstream*." Saber, ¶200.

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.[7]

[7] As discussed *supra*, the reference to "VCEG-N55" in the '818 patent should read "VCEG-N5**2.**"



VCEG-N52, Fig. 1.

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, and doing so would involve only routine software engineering tasks with predictable results. Section VII.B.3; Saber, ¶¶201–02.

b. 1[a]

VCEG-N52/Sriram teaches 1[a]. Saber, ¶¶203–05. As described above, the applicant for the '818 patent acknowledged that VCEG-N52 disclosed the parameter set concept. VCEG-N52 discloses that "the coder **may generate new parameter sets** and convey them to the system. Parameter sets are transmitted out of band,

typically through a reliable control channel." VCEG-N52, 3. Further, VCEG-N52 discloses the "[e]ncoder and decoder maintain a synchronized set of parameter sets, which contain all information associated with a picture and/or **higher syntax layers**." VCEG-N52, 8. VCEG-N52 also discloses "int MaxPicId" under the Sequence Parameters heading (reproduced below). VCEG-N52, 8. Saber, ¶203.

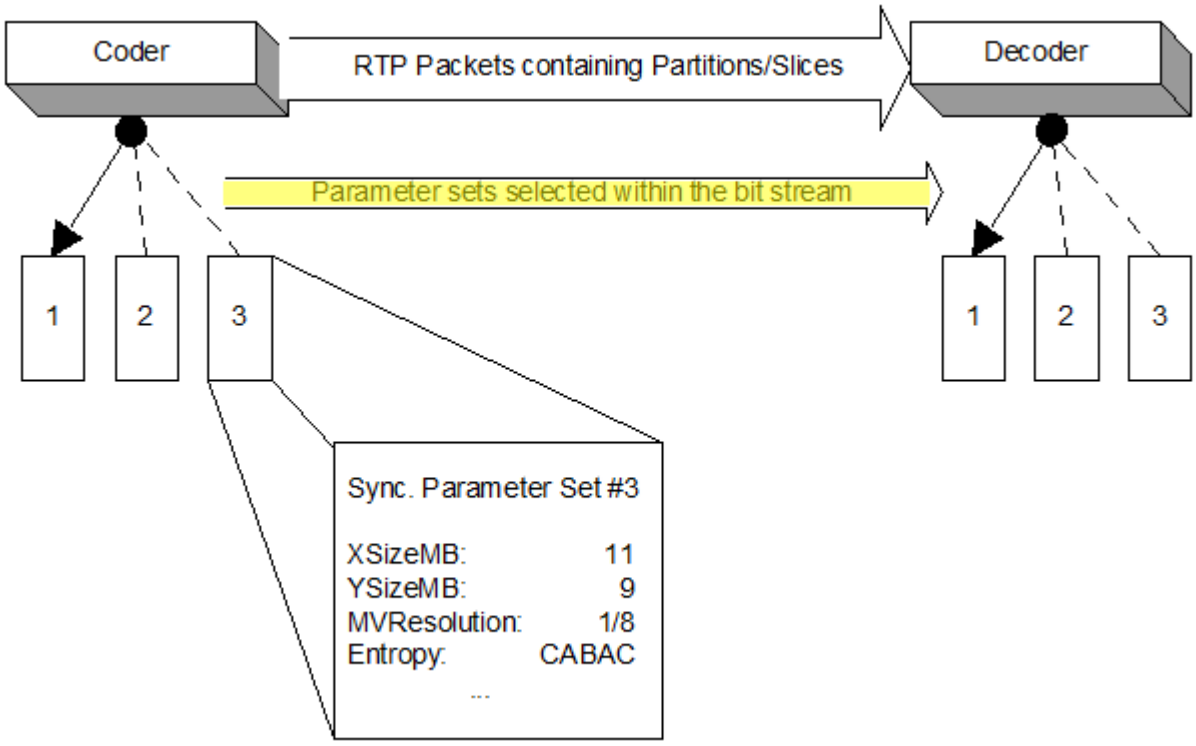
Sequence Parameters

int MaxPicId	Maximum number for the PicId Calculation. Is typically smaller than or equal to the number of reference pictures.
float PixAspectRatio	Pixel geometry, calculated by SizeX / SizeY, 0.0 indicates "unspecified"

Sriram discloses that "[e]ach video encoder 22 produces an encoded video signal which is applied to a transmission channel 24." Sriram, 4:24–27; Saber, ¶203. Sriram discloses that "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) 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 that Sriram suggests a sequence parameter data structure storing sequence parameters and a GOP parameter data structure storing GOP parameters, with the combination including VCEG-N52's "Sequence Parameters" and "Group of Picture Parameters." VCEG-N52, 8; Saber, ¶203. A POSITA would have been motivated to make this

combination (which would have been done using routine well-known and well-established software engineering techniques) 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, ¶203.

A POSITA would have understood that "*defining in the encoder*" encompasses an encoder creating the outgoing bitstream and identifying a "*sequence parameter set*" that would be incorporated into that bitstream. VCEG-N52 performs such "*defining*." Saber, ¶204. Figure 1 of VCEG-N52, reproduced below, shows "Parameter sets selected within the bit stream," sent from the "Coder" (encoder) to the "Decoder."



VCEG-N52, Fig. 1; Saber, ¶¶204–05.

c. 1[b]

VCEG-N52/Sriram teaches 1[b]. Saber, ¶¶206–08. As described above, the applicant for the '818 patent acknowledged that VCEG-N52 disclosed the parameter set concept. VCEG-N52 discloses the "[e]ncoder and decoder maintain a synchronized set of parameter sets, which contain all information associated with a picture and/or higher syntax layers" and "Picture Parameters," shown below. VCEG-N52, 8. Additionally, Sriram's picture parameter data structure includes multiple picture parameters, each of which teaches "at least one first picture parameter value," including, among other things, "horizontal_size," "vertical_size," and

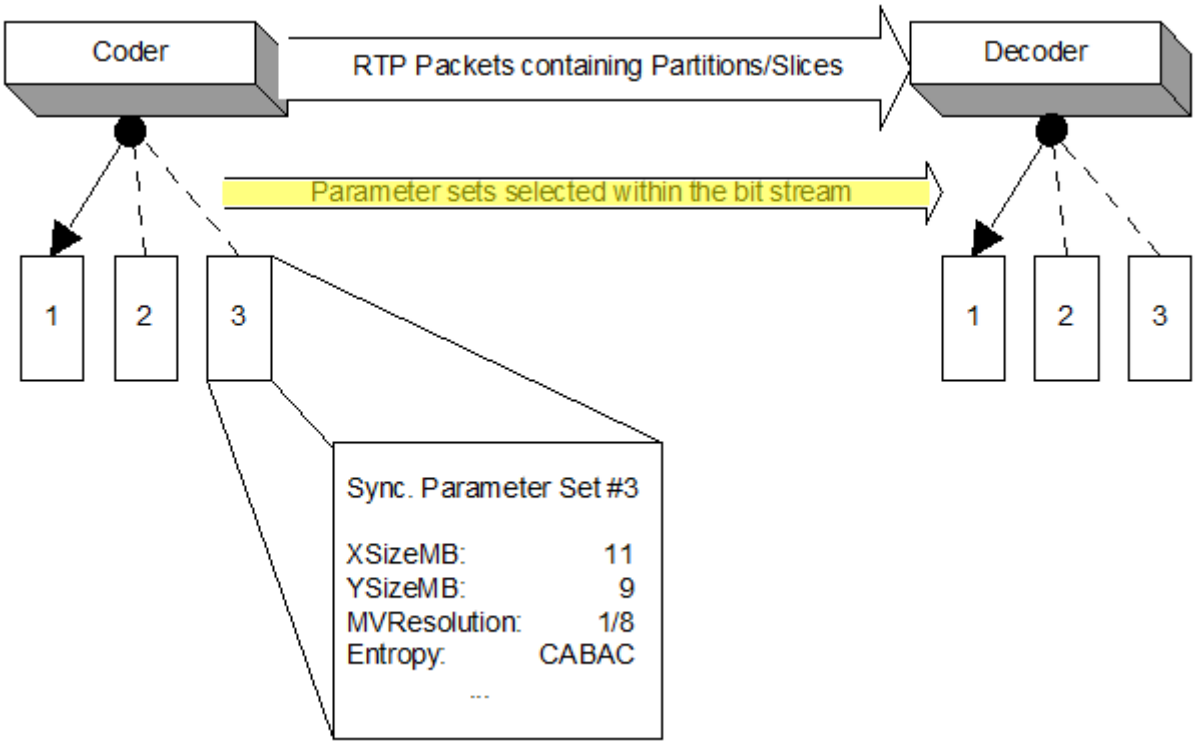
"chroma_format." Sriram, 14:35–54. A POSITA would have been motivated to combine VCEG-N52 with the picture parameters stored in Sriram's picture parameter data structure. Section VII.B.3; Saber, ¶206.

Picture Parameters

enum EntropyCoding	{UVLC, CABAC}
enum MotionResolution	{full, half, quarter, eigth}

VCEG-N52, 8 (blue highlight added).

A POSITA would have understood that "*defining in the encoder*" encompasses an encoder creating the outgoing bitstream and identifying a "*picture parameter set*" that would be incorporated into that bitstream. VCEG-N52 performs such "*defining*." Saber, ¶207. Figure 1 of VCEG-N52, reproduced below, shows "Parameter sets selected within the bit stream," sent from the "Coder" (encoder) to the "Decoder." Hence, a POSITA would have understood that the encoder in VCEG-N52 and Sriram defines "*parameter values in a picture parameter set for a picture*." *Id.*



VCEG-N52, Fig. 1 (yellow highlight added); Saber, ¶¶207–08.

d. 1[c]

VCEG-N52/Sriram teaches 1[c]. Saber, ¶¶209–12. As described above, the applicant for the '818 patent acknowledged that VCEG-N52 disclosed the parameter set concept. Both VCEG-N52 and Sriram disclose sending slice headers from the coder (encoder) to the decoder and extracting information from such slice headers. VCEG-N52, 4; Sriram, 1:47–49, 9:26–29; Saber, ¶209.

As to "*at least one picture parameter value in a slice header*," VCEG-N52 discloses "[t]he fundamental data structure of the video coder is a slice." VCEG-N52, 4. VCEG-N52 discloses that "[t]he pixel data of slice may be composed of a

maximum of three partitions. Partition A contains all header information including what is called in H.263 the slice header, motion vectors, MBTYPEs and similar." VCEG-N52, 4. VCEG-N52 further states that "[t]he Header Partition (Partition A) starts with a number of UVLC codewords that contain information about the slice" and disclose several parameters including "PictureID," which is "traditionally associated with the Slice header" and defined as a "symbol number [that] represents the PictureID of the picture the slice belongs to." VCEG-N52, 4–5; Saber, ¶210.

Sriram states that 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. A POSITA would have understood that for non-interlaced video, the majority of digital video, a picture and a frame are interchangeable terms. Saber, ¶211.

Therefore, a POSITA would have understood that the parameter "PictureID" would represent "*at least one picture parameter value in a slice header*," which "*remain[s] unchanged at least in all slice headers of one picture*" (in order for all the slices that belong to a given picture that is identified by PictureID to be properly encoded and subsequently decoded into that same picture that they belong to) and is "*defin[ed] in the encoder.*" Saber, ¶212.

5. Claim 2

VCEG-N52/Sriram teaches claim 2 for the same reasons as claim 7. *See*

Section VII.B.10; Saber, ¶213.

6. Claim 3

VCEG-N52/Sriram teaches claim 3 for the same reasons as claim 8. *See* Section VII.B.11; Saber, ¶214.

7. Claim 4

VCEG-N52/Sriram teaches claim 4 for the same reasons as claim 9. *See* Section VII.B.12; Saber, ¶215.

8. Claim 5

VCEG-N52 and Sriram teach claim 5. Saber, ¶¶216–18. Section VII.B.4 explains that the VCEG-N52/Sriram combination teaches the "method according to claim 1." As described above, the applicant for the '818 patent acknowledged that VCEG-N52 disclosed the parameter set concept. VCEG-N52 describes an advantage of "parameter sets" to include "making more transmissions (e.g., of picture headers) unnecessary" (VCEG-N52, 8), and that "[g]ood compression is possible in case of frequently changing picture parameters (such as picture size) once the parameters are available at the decoder." VCEG-N52, 8; Saber, ¶216.

VCEG-N52 discloses "the coder may generate new parameter sets and convey them to the system. Parameter sets are transmitted out of band, typically through a reliable control channel." VCEG-N52, 3. Further, VCEG-N52 discloses "[i]n many applications it will be sufficient to generate only a single parameter set, which will

be implicitly transmitted during capability exchange through the default operation point mechanism." VCEG-N52, 4. A POSITA would have understood that because the sequence parameter set and picture parameter set remain unchanged within a coded representation of a picture and may be the same set to be used in follow on picture(s), they would be transmitted "*less often than once per each picture*" for optimal compression. Saber, ¶¶217–18.

9. Claim 6

a. 6[pre]

If limiting, VCEG-N52 and Sriram teach 6[pre]. Saber, ¶¶219–21. VCEG-N52 and Sriram each disclose a video decoder. VCEG-N52, 4; Sriram, 4:11–12, Fig. 2. A POSITA would have understood that the combination would also include a decoder. Saber, ¶219.

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, ¶220.

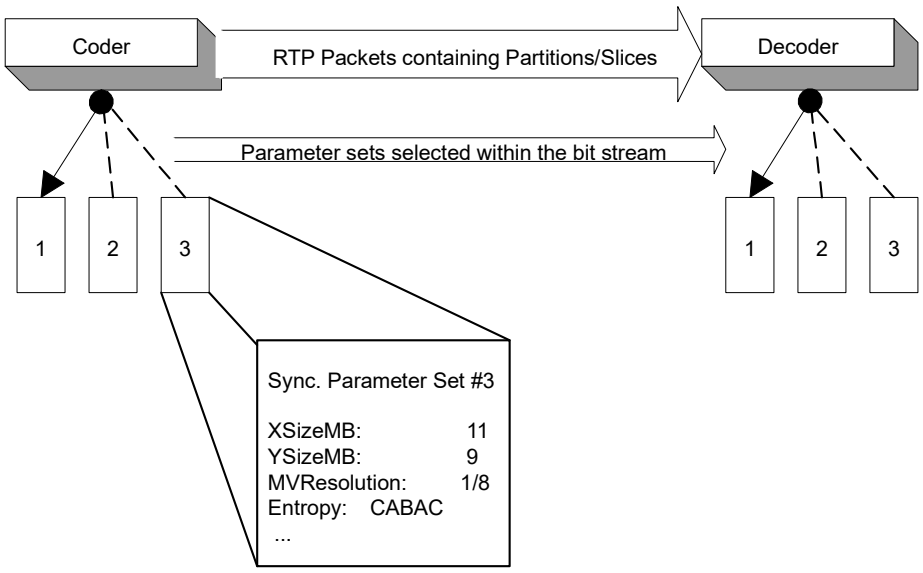
As to "*parameters are defined in a parameter set*," the '818 applicant admitted that VCEG-N52 discloses the parameter set concept. EX-1001, 2:49–58; EX-1002, 142; VCEG-N52, 3. A POSITA would have been motivated to combine VCEG-N52

with Sriram's teaching of different data structures for different levels of the video hierarchy. Section VII.B.3; Saber, ¶221.

b. 6[a]

VCEG-N52/Sriram teaches 6[a]. *Id*., ¶¶222–27. As described above, the '818 applicant acknowledged that VCEG-N52 disclosed the parameter set concept. EX-1001, 2:49–58; EX-1002, 142; VCEG-N52, 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. *See* Section VII.B.3; Section VII.B.4.b (discussion of VCEG-N52/Sriram disclosure of sequence and GOP levels); Saber, ¶222 (citing Sriram, 6:29–33, 12:64–13:1, 13:11–13, 17:55–60; VCEG-N52, 8).

A POSITA would have understood that "*recognizing, in the decoder*" encompasses a decoder parsing the incoming bitstream and identifying a "*sequence parameter set*." Saber, ¶223. VCEG-N52 performs such "*recognizing*." VCEG-N52's Figure 1, below, shows "Parameter sets selected within the bit stream," sent from the "Coder" (encoder) 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, ¶224.

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. Section VII.B.3; Saber, ¶225. The GOP parameter set and sequence parameter set are each "*a sequence parameter set.*" *Id*., ¶¶225–26 (citing EX-1001, 7:20–21, 7:54–57, 8:45–54, Fig. 1).

A POSITA would have understood the VCEG-N52/Sriram combination includes "*forming*" (e.g., creating in local memory) a sequence parameter. Saber, ¶¶226–27. In particular, Dr. Saber provides a syntax for the VCEG-N52/Sriram

combination, 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 that their associated values would be extracted by the decoder and stored in local memory for later use. *Id*.

c. 6[b]

VCEG-N52/Sriram teaches 6[b]. *Id*., ¶¶228–33. 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. A POSITA would have been motivated to combine VCEG-N52 with the picture parameters stored in Sriram's picture parameter data structure. Section VII.B.3; Saber, ¶228.

As discussed in 6[a], VCEG-N52 performs "*recognizing, in the decoder*." VCEG-N52, Figs. 1, 9–11; Section VII.B.9.b; Saber, ¶230.

In VCEG-N52/Sriram, the picture parameter set would include the parameters stored in Sriram's picture parameter data structure ("PicParam"). Sriram, 14:35–54; Saber, ¶231.

Further, 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. *See* Section VII.B.3; Sriram, 6:19–22; Saber, ¶232.

A POSITA would have further understood that the VCEG-N52/Sriram combination includes "*forming*" (e.g., creating in local memory) a picture parameter. Saber, ¶233. 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]

VCEG-N52/Sriram teaches 6[c]. Saber, ¶¶234–36. 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 that Sriram's slice data structure ("SliceData") includes both slice data and slice header information (e.g., a pointer to a data bit stream ("bitstream"), and variables that relate to the data (e.g., "VideoType," "PicFramePtr")). Sriram, 15:9–10, 15:52–66; Saber, ¶235. Although Sriram does not provide source code for the "PicFrame" structure to which PicFramePtr points, a POSITA would have understood from Sriram's disclosure

what such a structure would be. Saber, ¶235; *e.g.*, Sriram, 13:34–47, 14:26–34, 14:40–53, 14:63–15:7. A POSITA would have understood that the PicFrame would also have been defined using "typedef struct." Saber, ¶235. 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*. This would have included "the starting address of all the frames that are needed" for decoding. Sriram, 15:9–10; Saber, ¶235. A POSITA would have understood that 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, ¶235. 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 both 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 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*.; 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, ¶236. 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, ¶236.

Further, as discussed in Section VII.B.4.d (1[c]), 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, ¶236 (citing Sriram, 5:38–45; EX-1022, 19, 32–33).

e. 6[d]

VCEG-N52/Sriram teaches 6[d]. *Id*., ¶237. 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*.

10. Claim 7

VCEG-N52 and Sriram teach claim 7. Saber, ¶238. Section VII.B.9 explains that VCEG-N52/Sriram teaches the "method according to claim 6." As discussed in Section VII.B.9.d (6[c]), the combination teaches that 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, ¶238. As discussed in Section VII.B.9.c (6[b]), in VCEG-N52/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, ¶238.

11. Claim 8

VCEG-N52 and Sriram teach claim 8. Saber, ¶¶239–42. Section VII.B.9 explains that VCEG-N52/Sriram teaches the "method according to claim 6." As discussed in Sections VII.B.3 and VII.B.9.c (6[b]), in VCEG-N52/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, ¶239.

"GOPParameterSetIdentifier" is a reference to a GOP parameter set. Saber, ¶240. As discussed in Section VII.B.9.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 VCEG-N52/Sriram combination includes a reference to a sequence parameter set. *Id.*

Alternatively, as discussed in Section VII.B.3, in the VCEG-N52/Sriram combination, 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*. ¶241. A POSITA would have understood that claim 8's including "*a reference to a sequence parameter set*" encompasses such an indirect reference. *Id*.

If 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 VCEG-N52/Sriram combination to include a direct reference. *Id.*, ¶241. Sriram teaches that "[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, ¶242.

12. Claim 9

VCEG-N52 and Sriram teach claim 9. Saber, ¶243. Section VII.B.9 explains that the VCEG-N52/Sriram combination teaches the "method according to claim 6." As discussed in Sections VII.B.9.d (6[c]) and VII.B.9.e (6[d]), in the VCEG-N52/Sriram combination, 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. Sriram, 15:52–66; Saber, ¶243.

13. Claim 10

a. 10[pre]

VCEG-N52/Sriram teaches 10[pre] for the same reasons as 1[pre]. Section VII.B.4.a; Saber, ¶244.

b. 10[a]

VCEG-N52/Sriram teaches 10[a] for the same reasons as 11[a]. *See* Section VII.B.14.b; Saber, ¶¶245–48.

VCEG-N52 discloses a coder and decoder connected via an IP network. VCEG-N52, 1, 2. A POSITA would have understood that such a coder (encoder) would have been implemented using a processor. Saber, ¶246.

Sriram's "FIG. 1 [shown below] illustrate[s] a video decoding environment

20[.]" Saber, ¶247. In particular, "FIG. 1 illustrates…a set of video encoders 22A-22N. Each video encoder 22 produces an encoded video signal which is applied to a transmission channel 24, which may be any wire or wireless transmission channel." Sriram, 4:21–27.

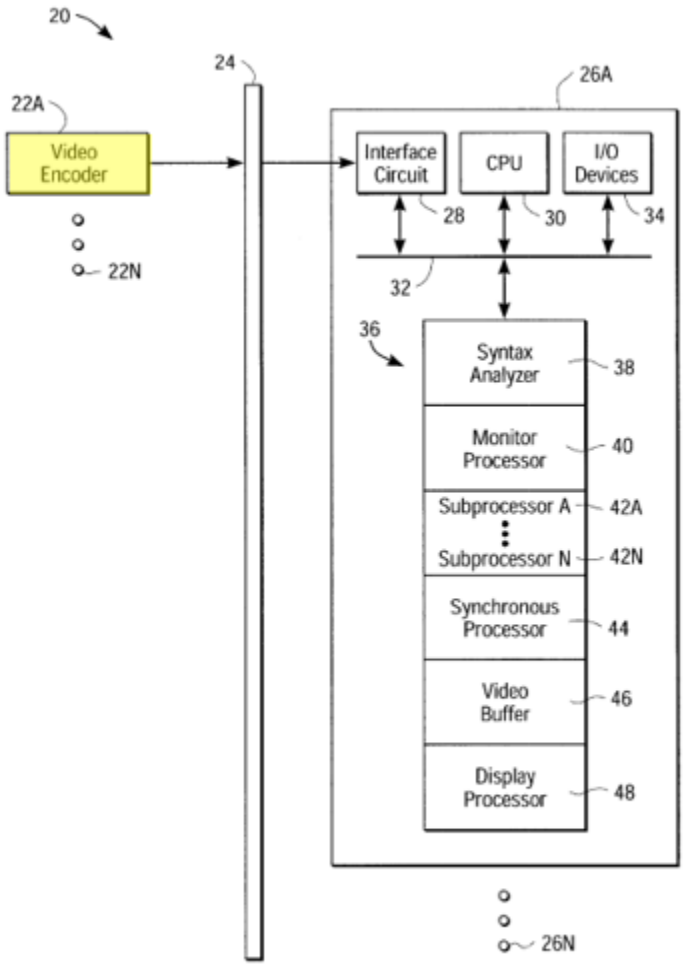


FIG. 1

Sriram, Fig. 1; Saber, ¶247.

In the VCEG-N52/Sriram combination, a POSITA would have understood that a processor performs the steps in 10[b]–10[d] as discussed below. Saber, ¶248.

c. 10[b]

VCEG-N52/Sriram teaches 10[b] for the same reasons as 1[a]. Section VII.B.4.b; Saber, ¶249.

d. 10[c]

VCEG-N52/Sriram teaches 10[c] for the same reasons as 1[b]. Section VII.B.4.c; Saber, ¶250.

e. 10[d]

VCEG-N52/Sriram teaches 10[d] for the same reasons as 1[c]. Section VII.B.4.d; Saber, ¶251.

14. Claim 11

a. 11[pre]

If limiting, combining VCEG-N52 and Sriram teaches 11[pre]. Saber, ¶252. Beyond 6[pre] (Section VII.B.9.a), both VCEG-N52 and Sriram disclose that such decoding is done by a decoder. VCEG-N52, 4; Sriram, 4:11–12, Fig. 2; Saber, ¶252.

b. 11[a]

VCEG-N52/Sriram teaches 11[a]. Saber, ¶¶253–56.

VCEG-N52 discloses a coder and decoder connected via an IP network. VCEG-N52, 1, 2. A POSITA would have understood that such a decoder would have been implemented using a processor. Saber, ¶254.

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, ¶255. The syntax analyzer program "extract[s] information from the video bit stream." Sriram, 5:17–19.

In the VCEG-N52/Sriram combination, a POSITA would have understood that a processor performs the steps in 11[b]–11[e] as discussed below. Saber, ¶256.

c. 11[b]

VCEG-N52/Sriram teaches 11[b] for the same reasons as 6[a]. Section VII.B.4.b; Saber, ¶257. 11[b] recites "*recognizing a sequence parameter set for a sequence of pictures [and] using the parameter set for a picture*." 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, ¶257. A POSITA would have understood that a group of pictures is a sequence of pictures and that 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]

VCEG-N52/Sriram teaches 11[c] for the same reasons as 6[b]. Section VII.B.9.c; Saber, ¶258.

e. 11[d]

VCEG-N52/Sriram teaches 11[d] for the same reasons as 6[c]. Section VII.B.9.d; Saber, ¶259.

f. 11[e]

VCEG-N52/Sriram teaches 11[e] for the same reasons as 6[d]. Section VII.B.9.e; Saber, ¶260.

15. Claim 12

a. 12[pre]

If limiting, combining VCEG-N52 and Sriram teaches 12[pre]. Saber, ¶¶261–63.

As discussed in Section VII.B.4.a (1[pre]), VCEG-N52/Sriram teaches "*[a] system comprising an encoder for encoding sequences of pictures into a bitstream*," and "*wherein parameters are defined in a parameter set and each picture comprises information of one or more slices*." Saber, ¶262.

As discussed in Section VII.B.14.a (11[pre]), VCEG-N52/Sriram teaches "*a decoder for decoding sequences of pictures from the bitstream*." Saber, ¶263.

b. 12[a]

VCEG-N52/Sriram teaches 12[a] for the same reasons as 10[a] and 11[a]. *See* Sections VII.B.13.b, VII.B.14.b; Saber, ¶264.

c. 12[b]

VCEG-N52/Sriram teaches 12[b] for the same reasons as 1[a]. Section VII.B.4.b; Saber, ¶265.

d. 12[c]

VCEG-N52/Sriram teaches 12[c] for the same reasons as 1[b]. Section VII.B.4.c; Saber, ¶266.

e. 12[d]

VCEG-N52/Sriram teaches 12[d] for the same reasons as 1[c]. Section VII.B.4.d; Saber, ¶267.

f. 12[e]

VCEG-N52/ Sriram teaches 12[e] for the same reasons as 6[pre]. Section VII.B.9.a; Saber, ¶268.

g. 12[f]

VCEG-N52/Sriram teaches 12[f] for the same reasons as 11[a]. Section VII.B.14.b; Saber, ¶269.

h. 12[g]

VCEG-N52/Sriram teaches 12[g] for the same reasons discussed in 6[a] and 11[b]. Sections VII.B.9.b, VII.B.14.c; Saber, ¶270.

i. 12[h]

VCEG-N52/Sriram teaches 12[h] for the same reasons as discussed in 6[b] and 15[c]. Sections VII.B.9.c, VII.B.18.d; Saber, ¶271.

j. 12[i]

VCEG-N52/Sriram teaches 12[i] for the same reasons in 6[c] and 11[d]. Sections VII.B.9.d, VII.B.14.e; Saber, ¶272.

k. 12[j]

VCEG-N52/Sriram teaches 12[j] for the same reasons as discussed in 6[d] and 11[e]. Sections VII.B.9.e, VII.B.14.f; Saber, ¶273.

16. Claim 13

VCEG-N52/Sriram teaches claim 13. Saber, ¶¶274–77. Section VII.B.15 explains that the VCEG-N52/Sriram combination teaches the "method according to claim 12." VCEG-N52 discloses "[t]he coder [encoder] also generates side information such as the start MB address and the number of MBs of the slice and the parameter set to be used." VCEG-N52, 3. VCEG-N52 discloses "the coder may generate new parameter sets and convey them to the system." VCEG-N52, 3; Saber, ¶275. "The coder can use parameter sets as soon as the system informs the coder that

they are available at the decoder…as an input, the coder has information about the available parameter sets at the decoder." VCEG-N52, 3. "Encoder and decoder maintain a synchronized set of parameter sets, which contain all information associated with a picture and/or higher syntax layers." VCEG-N52, 8. Therefore, VCEG-N52 teaches that a parameter set is stored in both the encoder and the decoder. Saber, ¶275.

Sriram also discloses that "[t]he video decoder 26 also includes a system memory 36 connected to the system bus 32." Saber, ¶276. And that "[t]he system memory 36, which may be primary and/or secondary memory, stores a set of executable programs utilized to implement the video decoding processes of the invention." Sriram, 4:42–46. A POSITA would have understood that the system memory would also provide the appropriate storage for the parameter set in order for the encoders and decoders to be able to function. Saber, ¶¶276–77.

17. Claim 14

a. 14[pre]

If limiting, VCEG-N52/Sriram teaches 14[pre] for the same reasons as 1[pre]. Section VII.B.4.a; Saber, ¶278.

b. 14[a]

VCEG-N52/Sriram teaches 14[a] for the same reasons as 10[a]. Section VII.B.14.b; Saber, ¶279.

c. 14[b]

VCEG-N52/Sriram teaches 14[b] for the same reasons as 1[a]. Section VII.B.4.b; Saber, ¶280.

d. 14[c]

VCEG-N52/Sriram teaches 14[c] for the same reasons as 1[b]. Section VII.B.4.c; Saber, ¶281.

e. 14[d]

VCEG-N52/Sriram teaches 14[d] for the same reasons as 1[c]. Section VII.B.4.d; Saber, ¶282.

18. Claim 15

a. 15[pre]

If limiting, combining VCEG-N52 and Sriram teaches 15[pre]. Saber, ¶283. 15[pre] adds only a "*receiving device*" comprising such a decoder to 11[pre] (Section VII.B.14.a). Both VCEG-N52 and Sriram disclose that 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. Similarly, Sriram uses its invention in workstations, and its decoder is part of a system that is connected to input/output devices. Sriram, 3:54–59, 4:34–41, 17:36–39.

b. 15[a]

VCEG-N52/Sriram teaches 15[a] for the same reasons as 11[a]. Section VII.B.14.b; Saber, ¶284.

c. 15[b]

VCEG-N52/Sriram teaches 15[b] for the same reasons as 11[b]. Section VII.B.14.c; Saber, ¶285.

d. 15[c]

VCEG-N52/Sriram teaches 15[c] for the same reasons as 11[c]. Section VII.B.14.d; Saber, ¶286.

e. 15[d]

VCEG-N52/Sriram teaches 15[d] for the same reasons as 11[d]. Section VII.B.14.e; Saber, ¶287.

f. 15[e]

VCEG-N52/Sriram teaches 15[e] for the same reasons as 11[e]. Section VII.B.14.f; Saber, ¶288.

19. Claim 16

a. 16[pre]

If limiting, VCEG-N52/Sriram teaches 16[pre] for the same reasons as 1[pre]. Section VII.B.4.a; Saber, ¶289.

b. 16[a]

VCEG-N52/Sriram teaches 16[a] for the same reasons as 1[pre]. Section VII.B.4.a; Saber, ¶290.

c. 16[b]

VCEG-N52/Sriram teaches 16[b] for the same reasons as 1[a]. Section VII.B.4.b; Saber, ¶291.

d. 16[c]

VCEG-N52/Sriram teaches 16[c] for the same reasons as 1[b]. Section VII.B.4.c; Saber, ¶292.

e. 16[d]

VCEG-N52/Sriram teaches 16[d] for the same reasons as 1[c]. Section VII.B.4.d; Saber, ¶293.

f. 16[e]

VCEG-N52/Sriram teaches 16[e] for the same reasons as 1[c]. Section VII.B.4.d; Saber, ¶294.

20. Claim 17

VCEG-N52/Sriram teaches claim 17 for the same reasons as claim 7. Section VII.B.10; Saber, ¶295.

21. Claim 18

VCEG-N52/Sriram teaches claim 18 for the same reasons as claim 8. Section

VII.B.11; Saber, ¶296.

22. Claim 19

VCEG-N52/Sriram teaches claim 19 for the same reasons as claim 9. Section VII.B.12; Saber, ¶297.

23. Claim 20

a. 20[pre]

If limiting, combining VCEG-N52 and Sriram teaches 20[pre]. Saber, ¶¶298–301.

As discussed in Section VII.B. 4.a (1[pre]), the VCEG-N52/Sriram combination teaches "*encoding sequences of pictures into a bitstream, wherein parameters are defined in a parameter set and each picture comprises information of one or more slices*." Section VII.B.13.b (10[a]) explains that this combination teaches a processor configured to perform the recited steps that follow. VCEG-N52/Sriram discloses the additional requirement of a "*computer readable storage medium embedded with a computer program comprising programming code for execution on a processor*," and that program is for the "*encoding.*" Saber, ¶299.

As discussed in Section VII.B.13.a (10[pre]) and Section VII.B.4.a (1[pre]), both VCEG-N52 and Sriram disclose encoding done by an encoder. VCEG-N52, 3, Fig. 1; Sriram, 4:21–27. And as discussed in Section VII.B.13.b (10[a]), a POSITA would have understood that both references use a processor for encoding a video and

would have been implemented with programming code in a computer program stored on a computer readable storage medium. Saber, ¶¶300–01.

Sriram describes "system memory" ("*a computer readable storage medium*") that "stores a set of executable programs" ("*programming code*") "utilized to implement the video decoding process of the invention." Sriram, 4:34–46. The system memory 36 is connected to the central processing unit via a system bus. *Id.*, 4:34–38, Fig. 1. A POSITA would have understood that the central processing unit ("*a processor*") could access and execute the programs stored in system memory. Saber, ¶301.

b. 20[a]

VCEG-N52/Sriram teaches 20[a] for the same reasons as 1[a]. Section VII.B.4.b; Saber, ¶302.

c. 20[b]

VCEG-N52/Sriram teaches 20[b] for the same reasons as 1[b]. Section VII.B.4.c; Saber, ¶303.

d. 20[c]

VCEG-N52/Sriram teaches 20[c] for the same reasons as 1[c]. Section VII.B.4.d; Saber, ¶304.

24. Claim 21

a. 21[pre]

If limiting, combining VCEG-N52 and Sriram teaches 21[pre]. Saber, ¶¶305–07. Beyond 6[pre] (Section VII.B.9.a) and 11[a] (Section VII.B.14.b), VCEG-N52/Sriram discloses the additional "*computer readable storage medium embedded with a computer program comprising programming code for execution on a processor*," and that program is for the "*decoding.*" *Id*., ¶¶305–06.

As discussed in Section VII.B.23.a (20[pre]), a POSITA would have understood that the connection between the central processing unit and the system memory ("*a computer readable storage medium*") would be so the central processing unit ("*a processor*") could access and execute the programs ("*programming code*") stored in system memory. Saber, ¶307 (citing Sriram, 4:34–46, Fig. 1).

b. 21[a]

VCEG-N52/Sriram teaches 21[a] for the same reasons as 11[b]. Section VII.B.14.c; Saber, ¶308.

c. 21[b]

VCEG-N52/Sriram teaches 21[b] for the same reasons as 6[b]. Section VII.B.9.c; Saber, ¶309.

d. 21[c]

VCEG-N52/Sriram teaches 21[c] for the same reasons as 6[c]. Section VII.B.9.d; Saber, ¶310.

e. 21[d]

VCEG-N52/Sriram teaches 21[d] for the same reasons as 6[d]. Section VII.B.9.e; Saber, ¶311.

25. Claim 22

a. 22[pre]

If limiting, VCEG-N52/Sriram teaches 22[pre] for the same reasons as 1[pre]. Section VII.B.4.a; Saber, ¶312.

b. 22[a]

VCEG-N52/Sriram teaches 22[a] regardless of whether it is construed as MPF. Saber, ¶313. VCEG-N52/Sriram teaches 22[a] for the same reasons as 1[a]. *Id.*; Section VII.B.4.b. As shown below, Figure 3 of the ’818 patent discloses a structure that includes a processor, encoding buffer, and program code. *See also, e.g.*, Sriram, 4:42–46.

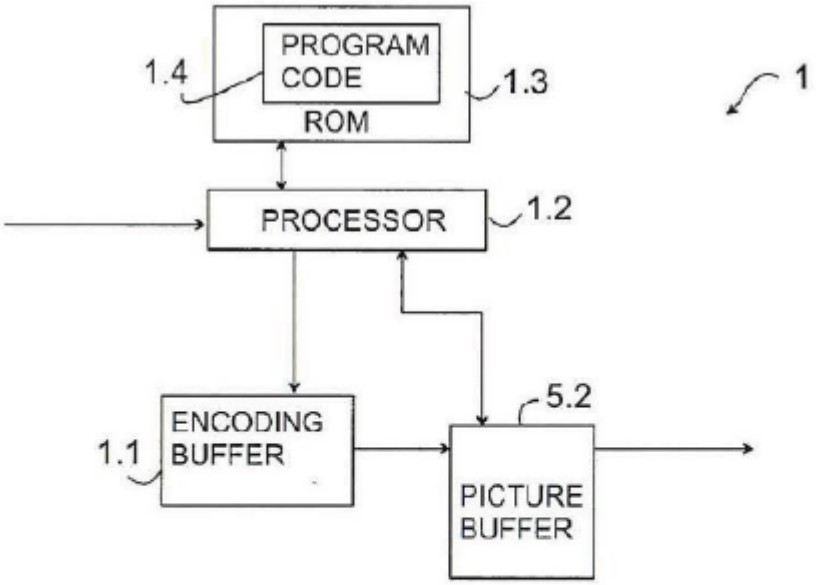


Fig. 3

As discussed in Section VII.B.4.b (1[a]), VCEG-N52/Sriram discloses "*defining, in an encoder, parameter values in a sequence parameter set for a sequence of pictures*." Saber, ¶314. Section VII.B.23.a (20[pre]) explains that 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. Additionally, VCEG-N52 discloses that "[s]upport of retransmission is required in **encoders, with a minimum AL3 SentBufferSize** of 1024 octets." VCEG-N52, 13. Sriram further teaches "video

frames…buffered in **memory**." Sriram, 2:52–55; Saber, ¶¶314–15.

c. 22[b]

VCEG-N52/Sriram teaches 22[b] regardless of whether it is construed as MPF. Saber, ¶316. VCEG-N52/Sriram teaches 22[b] for the same reasons as 1[b]. Section VII.B.4.c; Saber, ¶316. If this term is construed as MPF, VCEG-N52/Sriram's encoder teaches the structure discussed in Section VIII.A ("an encoder as described in the '818 patent specification in figure 3 and equivalents thereof") for the same reasons discussed in Section VII.B.25.b (22[a]) above, and that encoder performs the recited function in 22[b]. Saber, ¶316.

d. 22[c]

VCEG-N52/Sriram teaches 22[c] regardless of whether it is construed as MPF. Saber, ¶317. VCEG-N52/Sriram teaches 22[c] for the same reasons as 1[c]. *See* Section VII.B.4.d; Saber, ¶317. If this term is construed as MPF, VCEG-N52/Sriram's decoder teaches the structure discussed in Section VIII.A ("an encoder as described in the '818 patent specification in figure 3 and equivalents thereof") for the same reasons discussed in Section VII.B.25.b (22[a]) above, and that encoder performs the recited function in 22[c]. Saber, ¶317.

26. Claim 23

a. 23[pre]

If limiting, VCEG-N52/Sriram teaches 23[pre] for the same reasons as 11[pre]. Section VII.B.14.a; Saber, ¶318.

b. 23[a]

VCEG-N52/Sriram teaches 23[a] regardless of whether it is construed as MPF. Saber, ¶¶319–22. The VCEG-N52/Sriram combination teaches the function as discussed in Section VII.B.14.c (11[b]). *Id.*, ¶237. If 23[a] is construed as MPF, the combination teaches the structure discussed in Section VI.A ("a decoder as described in the '818 patent specification in figure 4 and equivalents thereof"). Saber, ¶321. As shown below, '818 patent Figure 4 discloses a structure including a processor, decoding buffer, and program code (decoding buffer and program code both stored in memory).

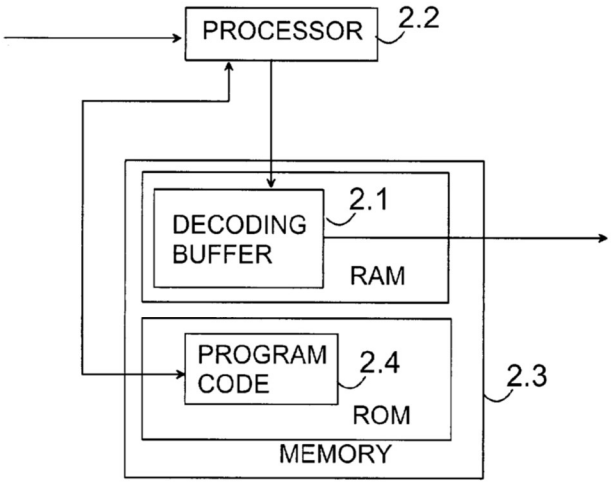


Fig. 4

As discussed in Section VII.B.14.c (11[b]), the VCEG-N52/Sriram combination teaches that its decoding is performed by a decoder. VCEG-N52, 1, 2; Sriram, 4:34–58, Figs. 1, 2; Saber, ¶322. Section VII.B.14.b (11[a]) explains that this combination teaches a processor configured to perform the recited steps that follow. Section VII.B.24.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, ¶322. In particular, Sriram discloses storing executable programs in system memory for decoding. Sriram, 4:34–46. The VCEG-N52/Sriram combination 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 discloses "video frames" that are "buffered in memory." *Id.*, 2:52–55, Fig. 2. Therefore, a POSITA would have understood that the combination 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, ¶322.

c. 23[b]

VCEG-N52/Sriram teaches 23[b] regardless of whether it is construed as MPF. *Id.*, ¶¶323–25. VCEG-N52/Sriram teaches the function as discussed in Section VII.B.9.c (6[b]). *Id.*, ¶324. If 23[b] is construed as MPF, the combination 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, ¶325.

d. 23[c]

VCEG-N52/Sriram teaches 23[c] regardless of whether it is construed as MPF. *Id.*, ¶¶326–28. VCEG-N52/Sriram teaches the function as discussed in Section VII.B.9.d (6[c]). *Id.*, ¶327. If 23[c] is construed as MPF, the combination 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, ¶328.

e. 23[d]

VCEG-N52/Sriram teaches 23[d] regardless of whether it is construed as MPF. *Id.*, ¶¶329–31. VCEG-N52/Sriram teaches the function as discussed in Section VII.B.9.e (6[d]). *Id.*, ¶330. If 23[d] is construed as MPF, the combination 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, ¶331.

VIII. CONCLUSION

Inter Partes Review of the Challenged Claims is respectfully requested.

IX. MANDATORY NOTICES

A. Real Party in Interest

Petitioner identifies themselves as the real party in interest.

B. Related Matters

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

- *Element Television Company, LLC et al v. Nokia Corporation a/k/a Nokia of America Corporation et al.*, Case No. 0:24-cv-04269 (D.M.N. Nov. 25, 2024)
- *Amazon.com, Inc. et al v. Nokia Technologies Oy*, IPR2024-00572 (PTAB Apr. 1, 2024)
- *Nokia Technologies Oy v. Amazon.com, Inc. et al.*, 1-23-cv-01236 (D.D.E. Oct. 31, 2023)
- *Nokia Technologies Oy v. HP Inc. f/k/a Hewlett-Packard Company*, 1-23-cv-01237 (D.D.E. Oct. 31, 2023)
- *Certain Video Capable Electronic Devices, Including Computers, Streaming Devices, Televisions, and Components and Modules Thereof*, Inv. No. 337-TA-1380 (ITC Oct. 31, 2023)
- *Nokia Corporation et al v. Amazon.com, Inc.*, 1-23-cv-01232 (D.D.E. Oct. 27, 2023)
- *Certain Electronic Devices, Including Computers, Tablet Computers, and Components and Modules Thereof*, 337-TA-1208 (ITC Jul. 1, 2020)
- *Nokia Technologies Oy v. Lenovo Group Ltd., et al.*, 5-19-cv-00427 (E.D.N.C. Sept. 25, 2019)

C. Notice of Counsel and Service Information

Lead Counsel	Back-Up Counsel
Brian E. Ferguson Reg. No. 36,801 WINSTON & STRAWN LLP 1901 L Street, N.W. Washington, D.C. 20036 T: 202-282-5276 beferguson@winston.com	Eimeric Reig (*pro hac vice* to be submitted) WINSTON & STRAWN LLP 101 California Street 21st Floor San Francisco, CA 94111 T: 415-591-6808 ereigplessis@winston.com
Back-Up Counsel	**Back-Up Counsel**
Joseph C. Masullo (*pro hac vice* to be submitted) WINSTON & STRAWN LLP 1901 L Street, N.W. Washington, D.C. 20036 T: 202-282-5897 jmasullo@winston.com	Lauren M. Rennecker (*pro hac vice* to be submitted) WINSTON & STRAWN LLP 1901 L Street, N.W. Washington, D.C. 20036 T: 202-282-5897 lrennecker@winston.com

Dated: July 23, 2025

Respectfully Submitted,

/s/ Brian E. Ferguson
Brian E. Ferguson
Reg. No. 36,801
WINSTON & STRAWN LLC
1901 L Street, N.W.
Washington, D.C. 20036
T: 202-282-5276
beferguson@winston.com

CERTIFICATE OF COMPLIANCE

I hereby certify that this brief complies with the type-volume limitations of 37 C.F.R. § 42.24, because it contains 13,981 words (as determined by the Microsoft Word word-processing system used to prepare the brief and including annotations to the figures), excluding the parts of the brief exempted by 37 C.F.R. § 42.24.

Dated: July 23, 2025

Respectfully Submitted,

/s/ Brian E. Ferguson

Brian E. Ferguson
Reg. No. 36,801
WINSTON & STRAWN LLC
1901 L Street, N.W.
Washington, D.C. 20036
T: 202-282-5276
beferguson@winston.com

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. § 42.6(e), I hereby certify that on this 23rd day of July, 2025, I caused to be served a true and correct copy of the foregoing and any accompanying exhibits by priority mail on the following:

Nokia Corporation and Alston & Bird LLP
Vantage South End
1120 South Tryon Street
Suite 300
Charlotte, NC 28203-6818

Dated: July 23, 2025

Respectfully Submitted,

/s/ Brian E. Ferguson
Brian E. Ferguson
Reg. No. 36,801
WINSTON & STRAWN LLC
1901 L Street, N.W.
Washington, D.C. 20036
T: 202-282-5276
beferguson@winston.com