UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ELEMENT TV COMPANY, LP, AND
ELEMENT TELEVISION COMPANY, LLC

Petitioners

v.

NOKIA CORPORATION AND NOKIA TECHNOLOGIES OY,

Patent Owner.

IPR2025-01068

U.S. Patent No. 8,050,321

PETITION FOR *INTER PARTES* REVIEW
OF U.S. PATENT NO. 8,050,321

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

TABLE OF CONTENTS

I. INTRODUCTION 1

II. GROUNDS FOR STANDING 1

III. IDENTIFICATION OF CHALLENGE 1

A. Prior Art 1

B. Grounds for Challenge 2

IV. '321 PATENT OVERVIEW 2

A. Technology Overview 2

B. The '321 Patent's Purported Invention 5

C. '321 Patent Prosecution History 7

V. LEVEL OF ORDINARY SKILL 8

VI. CLAIM CONSTRUCTION 8

VII. DETAILED EXPLANATION OF GROUNDS 10

A. Ground 1: MPEG-1 Renders Obvious Claims 8, 10, and 11 10

1. MPEG-1 10

2. Claim 8 18

3. Claim 10 31

4. Claim 11 35

B. Ground 2: Kim Renders Obvious Claims 8, 10, and 11 39

1. Kim 39

2. Claim 8 43

3. Claim 10 63

4. Claim 11 66

C. Ground 3: MPEG-1 and Yagasaki Render Obvious Claim 9 70

1. Yagasaki 70

2. Claim 9 71

3. Motivation to Combine MPEG-1 and Yagasaki 75

VIII. CONCLUSION 77

Nokia Exhibit 2012, p.2
ASUS v. Nokia
IPR2025-01279

TABLE OF CONTENTS
(continued)

Page

IX. MANDATORY NOTICES 77

A. Real Party in Interest 77

B. Related Matters 78

C. Notice of Counsel and Service Information 78

Nokia Exhibit 2012, p.3
ASUS v. Nokia
IPR2025-01279

EXHIBIT LIST

Exhibit	Reference
1001	U.S. Patent No. 8,050,321
1002	U.S. Patent No. 8,050,321 File History
1003	Declaration of Dr. Eli Saber
1004	Curriculum Vitae of Dr. Eli Saber
1005	ISO/IEC 11172-2 (1st ed.) Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s - Part 2: Video ("MPEG-1")
1006	U.S. Patent No. 6,912,351 ("Kim")
1007	CL480PC MPEG-1 System Decoder Specification, *available at* https://web.archive.org/web/19970128081644/http:/www.c-cube.com/prdctlst/cl480pc.pdf.
1008	U.S. Patent No. 7,263,270
1009	Ruby B. Lee et al., *Real-Time Software MPEG Video Decoder on Multimedia-Enhanced PA 7100LC Processors*, Hewlett-Packard J., Apr. 1995, *available at* https://archive.org/details/Hewlett-Packard_Journal_Vol._46_No._2_1995-04_Hewlett-Packard
1010	Ketan Patel, et al., *Performance of a Software MPEG Video Decoder*, Multimedia '93: Proceedings of the First ACM Int'l Conf. on Multimedia, August 2-6, 1993
1011	Complaint, *Element Television Co. v. Nokia Corp.*, No. 24-cv-04269 (D. Mn. Nov. 25, 2024), ECF No. 1
1012	RESERVED
1013	Declaration of June Ann Munford
1014	U.S. Patent No. 7,894,521
1015	U.S. Patent No. 5,786,858 ("Yagasaki")

LISTING OF CHALLENGED CLAIMS

Reference	Claim Limitation
Claim 8	
8[pre]	A method for decoding a compressed video sequence, the method comprising:
8[a]	decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence;
8[b]	starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame;
8[c]	decoding identifier values for image frames according to a numbering scheme; and
8[d]	resetting the identifier value for the indicated first image frame of the independent sequence.
Claim 9	
9	A method according to claim 8, wherein the indication is a separate flag included in the header of a slice.
Claim 10	
10[pre]	A video decoder comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the video decoder to decode a compressed video sequence, by causing the video decoder to at least:

Nokia Exhibit 2012, p.5
ASUS v. Nokia
IPR2025-01279

Reference	Claim Limitation
10[a]	decode from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence;
10[b]	start the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame; and
10[c]	decode identifier values for image frames according to a numbering scheme; and
10[d]	reset the identifier value for the indicated first image frame of the independent sequence.
Claim 11	
11[pre]	A computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence, the computer program product comprising:
11[a]	a computer program code for decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence;
11[b]	a computer program code for starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame;

Reference	Claim Limitation
11[c]	a computer program code for decoding identifier values for image frames according to a numbering scheme; and
11[d]	a computer program code for resetting the identifier value for the indicated first image frame of the independent sequence.

I. INTRODUCTION

Element TV Company, LP, and Element Television Company, LLC ("Element" or "Petitioners") request *inter partes* review ("IPR") of claims 8-11 ("Challenged Claims") of U.S. Patent 8,050,321 (EX1001 or "'321 patent").

II. GROUNDS FOR STANDING

Petitioners certify the '321 patent is available for IPR, and Petitioners are not barred or estopped from requesting IPR on the grounds herein. Petitioners file this Petition within one year of filing a district court complaint against Nokia Corporation and Nokia Technologies Oy (collectively, "Patent Owner" or "PO") seeking declaratory judgment of non-infringement of the '321 patent. *See Element Television Co. v. Nokia Corp.*, No. 24-cv-04269 (D. Mn.) (EX1011; EX1012).

III. IDENTIFICATION OF CHALLENGE

A. Prior Art

Petitioners respectfully request review and cancellation under 35 U.S.C. §311 of the Challenged Claims in view of:

Prior Art
"MPEG-1," ISO/IEC 11172-2 (1st ed.), published on August 1, 1993; prior art at least under §102(b)[1,2] (EX1005)
"Kim", U.S. Patent No. 6,912,351, filed on October 19, 1998; prior art at least under §102(e) (EX1006)
"Yagasaki," U.S. Patent No. 5,786,858, filed October 7, 1996; prior art at least under §102(b) (EX1013)

B. Grounds for Challenge

Ground		Claims	Prior Art
1	§103	8, 10, 11	MPEG-1
2	§103	8, 10, 11	Kim
3	§103	9	MPEG-1, Yagasaki

IV. '321 PATENT OVERVIEW

A. Technology Overview

The '321 patent generally concerns encoding and decoding video comprising independent sequences of image frames. (EX1001, Abstract.) As the title ("Grouping of Image Frames in Video Coding") suggests, a key concept discussed in the '321 patent is the grouping of these image frames. (EX1001.) The concept

[1] References to 35 U.S.C. §§102, 103 are to the pre-AIA statutory framework.

[2] MPEG-1 was publicly accessible in 1993, more than eight years before the earliest claimed priority date of the '321 patent (January 23, 2002). (EX1013, ¶9; EX1001, (60).)

Nokia Exhibit 2012, p.9
ASUS v. Nokia
IPR2025-01279

of grouping image frames into displayable sequences has existed since at least 1993, when International Standard ISO/IEC 11172-2 ("MPEG-1") was released. (EX1003, ¶48.) MPEG-1 used the term "group of pictures" or "GOP" to refer to a group of image frames. (EX1005, 23, 71.)[3] The '321 patent uses these terms in the same way. (*E.g.*, EX1001, 2:34-40.)

The '321 patent discusses the "independently decodable [GOP]" or "independent sequence" (EX1001, 11:26-30, Abstract), which contains pictures that can be decoded independently of information outside the GOP. (EX1003, ¶49.) Not all GOPs are independently decodable because image frames may depend on information from other frames, either within the same GOP or outside of it. (EX1001, 3:52-56.) This dependency arises from the use of a compression technique called "temporal prediction," which takes advantage of "temporal redundancy" across image frames.

Temporal redundancy concerns similarities between image frames. (EX1001, 2:1-3.) Because image frames in a video sequence are typically captured in quick succession, many aspects of the captured scene do not change or change in

[3] Page numbers for non-patent references refer to the numbers added by Petitioners.

Nokia Exhibit 2012, p.10
ASUS v. Nokia
IPR2025-01279

predictable ways from one image frame to the next. (EX1003, ¶50.) These aspects are considered temporally redundant.

Temporal prediction uses information about one frame to predict information about another frame from a different time and takes advantage of temporal redundancies by encoding only the changes that occur between image frames. Temporal prediction allows image frames to be encoded using less data because encoding does not have to include redundant information. Instead, the encoded information contains only some information about the depicted scene and refers to another image frame for any missing information. (EX1003, ¶¶50-51.)

A specific type of temporal prediction mentioned in the '321 patent is "motion-compensated temporal prediction," which predicts the location of an object in an image frame based on its location in other image frames and information about its movement. (EX1001, 2:18-24.)

When an image frame is encoded using temporal prediction, additional information outside of the encoded image frame is needed to decode it, namely, the reference image frame on which it depends. (EX1003, ¶¶52-53; EX1001, 2:31-33.)

There are different "types" of image frames based on their dependence on other image frames. (*E.g.*, EX1001, 3:50-52; EX1003, ¶54.) The '321 patent discusses three types. "Intra-frames" or "I-frames" do not depend on information from other image frames. (EX1001, 2:23-29.) "Inter-frames," or "P-frames,"

Nokia Exhibit 2012, p.11
ASUS v. Nokia
IPR2025-01279

depend on information from previous I- or P-frames. (EX1001, 2:27-31.) "Bi-directionally predicted frames," or "B-frames," depend on both a preceding and succeeding reference frame. (EX1001, 2:43-48.)

As mentioned above, an independently decodable GOP or independent sequence is one that can be decoded without information from image frames in another GOP. In other words, in an independently decodable GOP, any image frames encoded using temporal prediction depend only on reference frames within the independently decodable GOP. (EX1003, ¶55.)

Independently decodable GOPs existed before the '321 patent. (EX1003, ¶56.) As MPEG-1 explained in 1993, "[t]he original concept of a group of pictures was a set of pictures that could be coded and displayed independently of any other group." (EX1005, 85.) MPEG-1 described ways to encode GOPs to ensure that they were independently decodable and still compliant with the MPEG-1 standard. (*E.g.*, EX1005, 85.)

B. The '321 Patent's Purported Invention

The '321 patent specification describes a "scalable coding hierarchy" in which video data is coded using "scalability layers." (EX1001, 5:12-25 (describing figures using a "scalable coding hierarchy" or "scalability"); EX1001, 7:43-55 (discussing "scalability layers").) But the concept of "scalable coding" is not found in the '321

patent claims.[4] *Compare* EX1001, cls.1-11, *with* EX1014, cls.1-24 (referring to "scalable, compressed video sequence" and "scalability layers"). The claims are instead directed to a simpler concept: grouping and numbering image frames. (EX1003, ¶¶57-59.) According to the patent, grouping and numbering help resolve "[a] problem in the numbering of image frames [that] arises when [a] user wishes to start browsing a video file in the middle of a video sequence." (EX1001, 11:11-13.) The '321 patent describes the problem as follows: "When the browsing of a video file is resumed from a random point after previous browsing, discontinuity typically occurs in the image numbering," which "[t]he decoder typically interprets…as [an] unintentional loss of image frames." (EX1001, 11:20-25.)

The '321 patent says this can be "avoided in the decoder by defining an initiation image in an independently decodable Group of Pictures GOP that is activated at the random point of the video file, and the number of the initiation image is set at zero." (EX1001, 11:26-30.) In other words, "when activated at a random point, the decoder preferably sets the identifier of the first image frame, preferably

[4] In a separate proceeding, PO asserted that the '321 patent "is directed to" "random access in video coding." Patent Owner's Response, *Amazon.com, Inc. v. Nokia Techs. Oy*, IPR2024-00691, Paper 20, 7 (Dec. 18, 2024). That phrase does not appear in the '321 patent specification, nor is the concept reflected in the claims.

an I-frame, of the independently decodable sub-sequence at zero." (EX1001, 11:37-40.) This allows the decoder to "find the correct image frame that initiates the subsequence from the video sequence of image frames." (EX1001, 11:46-49.) The Challenged Claims therefore recite an independent video sequence decoded along with each frame's identifier value, starting with a first frame (identified by an indication), and where the identifier value for the first frame is reset.

C. '321 Patent Prosecution History

The '321 patent issued from U.S. Application 11/338,934 (filed January 25, 2006) and claims priority to a continuation application filed November 29, 2002, and a foreign application filed January 23, 2002.[5]

Yagasaki was cited by the Examiner in a non-final rejection dated March 4, 2011, as part of an obviousness combination with Apostolopoulos (U.S. Pat. No. 7,103,669) against then-pending claim 11. (EX1002, 750-51.) Then-pending claim 11 recited "[a] method according to claim 10, wherein the indication is a separate flag included in the header of a slice." (EX1002, 735.) The Examiner found "Yagasaki discloses putting a flag for indicating an intra-frame coded block in the

[5] The references in this Petition qualify as prior art, even accepting the January 23, 2002, priority date. Petitioners reserve the right to further address priority if PO asserts an earlier date.

slice header (column 10, line 65 – column 11, line 2)." (EX1002, 750.) The Applicant neither disputed the Examiner's statement nor distinguished Yagasaki. Instead, the Applicant amended claim 11's parent, claim 10, in the next response and argued that amended claim 10 distinguished Apostolopoulos. (EX1002, 770-778.) Pending claim 11 ultimately issued as claim 9. (EX1002, 806.)

MPEG-1 and Kim were not before the Examiner during prosecution. (EX1002.)

V. LEVEL OF ORDINARY SKILL

A person of ordinary skill in the art ("POSITA") at the time of the alleged invention of the '321 patent would have had a (1) Bachelor's degree in electrical engineering, computer engineering, computer science, or a comparable field of study such as physics, and (2) approximately two to three years of practical experience with video encoding/decoding. Additional experience can substitute for the level of education, and vice-versa. (EX1003, ¶¶24-26.)

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); *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc). Petitioners are unaware of any

"prior claim construction determination" for the '321 patent. *See* 37 C.F.R. §42.100(b).

Petitioners do not propose any terms for construction because no construction is necessary to resolve any disputes identified in this Petition.[6] *Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999).

In a prior proceeding before the Board, PO proposed to construe "indication" in challenged claim 8 to mean "information in an encoded bitstream that necessarily confirms something to a decoder." *See Amazon.com,* IPR2024-00691, Paper 20, 34. For support, PO cited only the accompanying declaration of its expert, Dr. Richardson, who analogized the claimed "indication" to a syntax element described in a document regarding the H.264 standard that was published many years after the priority date of the '321 patent. *See id.* (citing EX2004). Importantly, PO cited no support for its proposed construction in the patent itself, and with good reason: PO's proposal adds a conclusiveness requirement that is inconsistent with the claim language, which does not require the "indication" to confirm or inform a decoder of anything with certainty but only requires an indication "of at least one image frame"

[6] Petitioners reserve the right to address construction of any terms raised at trial or in the related district court action. Petitioners further reserve the right to raise other grounds unavailable in an IPR in a separate action.

(i.e., something showing that at least one image frame, including the first image frame in decoding order, is present). (EX1001, 23:51-24:2; *Amazon.com,* IPR2024-00691, EX2005, 77:20-80:12, 83:10-85:4, 87:18-88:22.) The Board should therefore reject PO's proposal. But even if the Board adopts PO's proposed construction, the Challenged Claims would still have been obvious to a POSITA reading the prior art, as explained below. (EX1003, ¶63.)

VII. DETAILED EXPLANATION OF GROUNDS

A. Ground 1: MPEG-1 Renders Obvious Claims 8, 10, and 11

1. MPEG-1

a. Overview

When International Standard ISO/IEC 11172 ("the Standard") was approved in 1993, it included four parts: Systems, Video, Audio, and Compliance Testing. (EX1005, 11.) ISO/IEC later published a fifth part, which included a software simulation.

Historically, the term "MPEG-1" referred to all parts of the Standard. (EX1003, ¶65.) In this Petition, Petitioners use MPEG-1 to refer to the second part of the Standard, which concerns video. (*See* EX1005, 11.)

MPEG-1 was developed by the Moving Pictures Expert Group in response to "the growing need for a common format for representing compressed video on various digital storage media such as CDs, DATs, Winchester disks and optical

drives." (EX1005, 12, 68.) It "specifies a syntax for a coded video bitstream," (EX1005, 16) and defines the process for decoding that bitstream (EX1005, 15; *see also* EX1005, 18 ("This part of ISO/IEC 11172 specifies the coded representation of video for digital storage media and specifies the decoding process.").)

b. Key Concepts

MPEG-1 discloses five key concepts relevant to Ground 1: picture type, picture ordering, GOPs, closed GOPs, and Temporal Reference.

(i) Picture Type (I, P, and B Pictures)

In MPEG-1, "[a] picture corresponds to a single frame of motion video or to a movie frame." (EX1005, 71.) A "picture" in MPEG-1 corresponds to the claimed *image frames* in the '321 patent.[7] (EX1001, 2:55-58 ("video files…comprise a great number of still image frames, which are displayed rapidly in succession…to create an impression of a moving image").) Indeed, the '321 patent interchangeably uses the terms "pictures" and "image frames." (*E.g.*, EX1001, 4:20-23 ("An indication of at least one image frame is encoded into the video sequence, which indicated image frame is the first picture, in decoding order, of the independent sequence.").)

MPEG-1 describes three picture types relevant to this Petition: I-pictures, P-pictures, and B-pictures. (EX1005, 71; EX1003, ¶¶61-66.) An "I-picture" (intra-

[7] Italicized text is claim text unless otherwise noted.

coded picture) is "[a] picture coded using information only from itself" (EX1005, 23) and therefore "without reference to any other pictures" (EX1005, 71). A "P-picture" (predictive-coded picture) is "[a] picture that is coded using motion compensated prediction from the past reference picture." (EX1005, 25.) "[T]he past reference picture" is defined to be the nearest adjacent I- or P- picture that occurs at an earlier time in display order. (EX1005, 25 (defining "past reference picture"), 26 (defining "reference picture"); *see also* EX1005, 71 ("P-pictures…are coded using motion compensation from a previous I or P-picture.").) And a "B-picture" (bidirectionally predictive-coded picture) is "coded using motion compensated prediction from a past and/or future reference picture." (EX1005, 20; *see also* EX1005, 71.)

Each picture has a header with information about the picture, including the coding type. (EX1005, 87-88.)

(ii) Picture Ordering

MPEG-1 describes two "picture orderings": "display order" and "bitstream order." (EX1005, 84.) Display order is "[t]he order in which the decoded pictures should be displayed." (EX1005, 22, 87.) Bitstream order is the order in which pictures are decoded. (EX1005, 32, 71; EX1003, ¶74.) MPEG-1's "bitstream order" reflects the same picture ordering as the '321 patent's "decoding order." (EX1003, ¶76.)

As explained in MPEG-1, picture orderings may be different because of picture dependencies. (EX1005, 71.) For example, MPEG-1 provides the following example of a picture sequence in display order:[8]

I	B	B	P	B	B	P	B	B	P	B	B	I	B	B	P	B	B	P
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18

Figure D.3 -- Typical sequence of pictures in display order

The same sequence in bitstream order would be:

I	P	B	B	P	B	B	P	B	B	I	B	B	P	B	B	P	B	B
0	3	1	2	6	4	5	9	7	8	12	10	11	15	13	14	18	16	17

Figure D.4 -- Typical sequence of pictures in bitstream order

MPEG-1 explains that "[b]ecause the B-pictures depend on the following (in display order) I or P-picture, the I or P-picture must be transmitted and decoded before the dependent B-pictures." (EX1005, 71.) For example, in Figure D.3, picture P3 is fourth in display order, but it is second in decoding order in Figure D.4. This is because pictures B1 and B2 depend on picture P3 to be decoded. (EX1003, ¶75.)

[8] Annotations to figures have been added by Petitioners unless otherwise noted.

(iii) Groups of Pictures ("GOPs")

A GOP is "[a] series of one or more coded pictures intended to assist random access." (EX1005, 23.) "[I]t is the smallest coding unit that can be independently decoded within a sequence, and consists of a header and some number of pictures." (EX1005, 71.)

Pictures in a GOP, in bitstream order, "must start with an I-picture and may be followed by any number of I, P or B-pictures in any order." (EX1005, 84.) Pictures in a GOP, in display order, are "contiguous" and "must begin…with an I or a B-picture, and must end with an I or a P-picture." (EX1005, 84.)

Because GOPs end with an I- or a P-picture in display order, no picture in a GOP may depend on a reference picture in a subsequent GOP for motion compensation prediction. (EX1003, ¶¶77-79.) Neither I- nor P-pictures may depend on reference pictures in a subsequent GOP because I-pictures "are coded without reference to any other pictures," and P-pictures "are coded using motion compensation from a **previous** I or P-picture." (EX1005, 71.)[9] And no B-picture in the GOP may refer to a picture beyond the last I- or P-picture because reference pictures are "the **nearest adjacent** I- or P-pictures to the current picture in display

[9] Emphasis added unless otherwise indicated.

Nokia Exhibit 2012, p.21
ASUS v. Nokia
IPR2025-01279

order." (EX1005, 26.) Since GOPs must end with an I- or a P-picture, any B-picture may not refer to a picture beyond that final picture.

(iv) Closed GOPs

MPEG-1 explains "[t]he original concept of a group of pictures was a set of pictures that could be coded and displayed independently of any other group." (EX1005, 85.) Although MPEG-1 does not require that every GOP be independently decodable, "encoders can still construct groups of pictures which are independent of one another." (EX1005, 85.)

MPEG-1 discloses the concept of a closed GOP, which is a "group of pictures [that] has been encoded without motion vectors pointing to the previous group of pictures." (EX1005, 43.) A closed GOP therefore "can be decoded without using decoded pictures of the **previous** group for motion compensation." (EX1005, 85.) Closed GOPs, like all GOPs, also can be decoded without using decoded pictures of a **subsequent** group for motion compensation. (EX1003, ¶¶80-81.) Thus, MPEG-1 closed GOPs are independently decodable sequences of pictures.

(v) Temporal Reference

As noted above, each picture has a header with certain information about the picture, including the "Temporal Reference"—a "number which can be used to define the order in which the pictures must be displayed." (EX1005, 87.) MPEG-1 specifies that "[t]he first picture, in display order, in each group must have Temporal

Reference equal to zero," and Temporal Reference is "incremented by one for each picture in the group." (EX1005, 87, 44.) Temporal Reference thus gives the display order of pictures. (EX1005, 44 ("The temporal-reference is assigned (in sequence) to the pictures in display order."); EX1003, ¶82.)

c. Exemplary Closed GOP

MPEG-1 provides multiple ways to "construct groups of pictures which are independent of one another" (i.e., closed GOPs). (EX1005, 85.) "One way to do this is to omit any B-pictures preceding the first I-picture." (EX1005, 84.) Thus, MPEG-1 provides an exemplary closed GOP that begins with an I-picture. (EX1003, ¶¶83-86.) As explained below, this exemplary closed GOP teaches or suggests the challenged claims of the '321 patent.

MPEG-1 presents multiple examples of GOPs encoded without B-pictures preceding the first I-picture. For example, the following figure shows "[a] typical example of a closed group [of pictures]" presented in display order:

I	B	B	**P**	B	B	**P**	B	B	**P**	B	B	**P**
0	1	2	3	4	5	6	7	8	9	10	11	12

(a) closed group

MPEG-1, Fig.D.18(a)

(EX1005, 86.)

A key feature of this exemplary closed GOP is a single I-picture must be the first picture of the GOP in both decoding order and display order and must have a

Temporal Reference of zero. It is first in decoding order because the first picture in decoding order of any GOP must be an I-picture. (EX1005, 84 ("*Property 1.* A group of pictures, in bitstream order, must start with an I-picture.") (emphasis in original).) It is first in display order because the first picture of a GOP in display order must be an I- or a B-picture, and the example specifies any B-pictures preceding the first I-picture have been omitted. (EX1005, 84 ("*Property 2*. Another property of a group of pictures is that it must begin, in display order, with an I or a B-picture.") (emphasis in original).) Because this I-picture is first in display order, it has a Temporal Reference of zero. (EX1005, 87 ("The first picture, in display order, in each group must have Temporal Reference equal to zero.").) Thus, the first picture in decoding order of every exemplary closed GOP (i.e., every GOP encoded with an I-picture as the first picture in display order) has a Temporal Reference of zero. (EX1003, ¶83.)

This is shown in the following MPEG-1 example:

Example (a) in display order	**I** 0	B 1	**P** 2	B 3	**P** 4
Example (a) in decoding order	**I** 0	**P** 2	B 1	**P** 4	B 3

MPEG-1, Fig.D.19(a)

("Examples of Groups of Pictures and Temporal References")

(EX1005, 87.) The top row is a series of pictures in an exemplary GOP in display order, and the bottom row is the same exemplary GOP in decoding order. The numbers in the figure indicate each picture's Temporal Reference value. The I-picture in the exemplary GOP is first in both decoding and display order and has a Temporal Reference of zero. (EX1005, 87.)

2. Claim 8

a. 8[pre]

To the extent limiting, MPEG-1 teaches or suggests 8[pre]. (EX1003, ¶¶87-90.)

MPEG-1 discloses a *compressed video sequence*. MPEG-1 "specifies a coded representation that can be used for compressing video sequences to bitrates around 1.5 Mbit/s." (EX1005, 12; *see also* EX1005, 68 (MPEG-1 "defines a format for compressed digital video").) This coded representation of a compressed video sequence is called a "coded video bitstream." (EX1005, 16 ("[MPEG-1] specifies a syntax for a coded video bitstream"), 21 (defining coded bitstream as "[a] coded representation of a series of one or more pictures as specified in [MPEG-1]"), 69 (MPEG-1 describes the "structure of an [MPEG-1] bitstream" and "the coding operations that compress the video").)

MPEG-1 further discloses *a method for decoding* this coded video bitstream (*compressed video sequence*). As MPEG-1 explains, "[t]o view…moving pictures,

the decoder accesses the [MPEG-1] bitstream, and decodes it as described in [Section] D.7." (EX1005, 69; *see also* EX1005, 15 ("The decoding process is defined by [MPEG-1]."), 22 ("decoding (process): The process defined in [MPEG-1] that reads an input coded bitstream and produces decoded pictures or audio samples."), 14 ("[MPEG-1] specifies the syntax and semantics of the bitstream and the signal processing in the decoder.").) Section D.7 of MPEG-1 discloses the processes for decoding MPEG-1 video. (EX1005, 120-22.)

b. 8[a]

MPEG-1 teaches or suggests 8[a]. (EX1003, ¶¶91-102, 108-09.) Below, the Petition first discusses how MPEG-1's closed GOP teaches the recited *independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence*. The Petition then discusses how MPEG-1's closed_gop flag is an *indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence*. Alternatively, the Petition discusses how a Temporal Reference of zero in MPEG-1 is an *indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence*. Either teaching, considered together with MPEG-1's closed GOP, discloses 8[a].

(i) ***an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence***

As explained in §VII.A.1.b(iv), a closed GOP in MPEG-1 is decoded without using decoded pictures of the previous or subsequent group for motion compensation. This is because a closed GOP is defined to be a "group of pictures [that] has been encoded without motion vectors pointing to the previous group of pictures." (EX1005, 43.) All closed GOPs "must end with an I or a P-picture" and therefore cannot depend on pictures from a subsequent group of pictures for motion compensation. (EX1005, 84, 23 (an I-picture is "[a] picture coded using information only from itself"), 25 (a P-picture is "[a] picture that is coded using motion compensation prediction from the past reference picture"), 32 ("The order of the pictures in the coded stream is the order in which the decoder processes them in normal playback" and "[t]he last coded picture, in display order, of a group of pictures is either an I-Picture or a P-Picture.").)

Accordingly, MPEG-1's closed GOP is *an independent sequence* (does not depend on other sequences), *wherein all motion vectors* ("motion-compensated temporal prediction references") *point only to image frames within that closed GOP and not to image frames outside of the closed GOP* ("refer only to image frames within said independent sequence"). (EX1003, ¶¶88-90.)

(ii) ***decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence***

As discussed above, MPEG-1's closed GOP is *an independent sequence*. MPEG-1's closed_gop flag teaches or suggests the rest of 8[a]. Alternatively, a Temporal Reference of zero teaches or suggests the rest of 8[a].

The closed_gop flag: In an MPEG-1 bitstream, "a group of pictures begins with a [GOP] header" that includes a "group_start_code," which "identifies the beginning of a group of pictures." (EX1005, 43, 85.)

The GOP header contains a closed_gop flag that indicates whether the GOP is open or closed. (EX1005, 43, 86.) And because a closed GOP is *an independent sequence*, a closed_gop flag indicating that the GOP is closed also indicates that the GOP is an independent sequence.

Furthermore, MPEG-1's GOP header precedes the first picture in the GOP in decoding order. (EX1005, 120 ("decoder will decode the group of pictures header, including the closed_gop and broken-link information" and then "decode the first picture header in the group of pictures").) The GOP header is thus *an indication* of *the first image frame in decoding order* of the GOP because it precedes that image frame.

And because the closed_gop flag is in the GOP header, the first image frame decoded after the closed_gop flag is the first image in decoding order of the GOP. (EX1005, 120 (decoding process for forward playback).) When the closed_gop flag indicates that the GOP is closed, it indicates that the next image frame decoded is *the first image frame in decoding order of an independent sequence*.

MPEG-1 decodes this indication because it discloses decoding the GOP header that includes the closed_gop flag. (EX1005, 120 ("decoder will decode the group of pictures header, including the closed_gop and broken-link information"); EX1003, ¶¶91-99.)

In a separate proceeding regarding the '321 patent, PO asserted the closed_gop flag is only an indication of whether the GOP is open or closed, not of a particular image frame, and because it is in the header, it is a syntax element that only applies to the entire GOP.[10] *Amazon.com,* IPR2024-00691, Paper 20, 32-33. But the closed_gop flag in the exemplary closed GOP **does** indicate that the next image frame decoded is the first one in decoding order of the independent sequence, and 8[a] does not require the *indication* to identify a particular image frame, only *at*

[10] Limitation 8[a], unlike claim 9, does not require the indication to be in a specific syntax layer.

least one image frame that is the first image frame in decoding order. (EX1003, ¶100.)

In addition, PO asserted that the closed_gop "does not confirm to a decoder any type of image as a first image frame"; rather, "an MPEG-1 decoder that receives a closed_gop flag with a value of '0' or '1' has no certainty about what frames follow in the bitstream." *Amazon.com,* IPR2024-00691, Paper 20, 34. This argument is based on PO's proposed construction for the term "indication" and should be rejected for the reasons discussed above in §VI. Nevertheless, even applying PO's construction, the closed_gop flag does confirm to the decoder that the next image frame decoded will be the first in decoding order in the exemplary closed GOP because, after decoding the closed_gop flag in the exemplary closed GOP header, the decoder recognizes that the next image frame it receives will be an I-picture that is the first image frame in decoding order of the independent sequence, as illustrated in MPEG-1 Example D.19(a).

Example (a) in display order	I 0	B 1	P 2	B 3	P 4
Example (a) in decoding order	I 0	P 2	B 1	P 4	B 3

(EX1005, 87; EX1003, ¶101.)

PO has further argued that MPEG-1's closed_gop flag does not support random access because it cannot indicate an "initiation picture" of a GOP where decoding can begin. *Amazon.com,* IPR2024-00691, Paper 20, 37. There is nothing in the claim, however, that requires support for random access.[11]

A Temporal Reference of zero: Alternatively, a Temporal Reference of zero in MPEG-1 teaches or suggests *decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of [the] independent sequence*.

As discussed in §VII.A.1.c, MPEG-1's exemplary closed GOP is encoded without B-pictures preceding the first I-picture in display order. As further explained in that Section, the first picture in decoding order of the exemplary closed GOP will always be an I-frame with a Temporal Reference of zero (because that I-

[11] PO additionally argued that MPEG-1's closed_gop flag cannot be the claimed *indication* based on closed GOPs **other than** the exemplary closed GOP. *Amazon.com*, IPR2024-00691, Paper 20, 37-40. But whether PO can conceive of an example that arguably does not practice the Challenged Claims is not the test for obviousness. *Bell Commc'ns Rsch., Inc. v. Vitalink Commc'ns Corp.,* 55 F.3d 615, 622-23 (Fed. Cir. 1995) (if a prior art reference sometimes, but not always, embodies a claim, it nonetheless teaches that claim).

Nokia Exhibit 2012, p.31
ASUS v. Nokia
IPR2025-01279

frame is the first frame in display order). Thus, in the exemplary closed GOP, the I-frame's Temporal Reference of zero is an *indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence*. And MPEG-1 discloses that the decoder decodes Temporal Reference. (EX1005, 120 (picture header decoded during forward playback); EX1003, ¶¶97-98.)

In a separate proceeding, PO argued that a Temporal Reference of zero indicates the first picture in display order, not necessarily in decoding order. *Amazon.com,* IPR2024-00691, Paper 20, 41. This again ignores the exemplary closed GOP in which an I-picture is the first image frame in both decoding and display order and, therefore, has a Temporal Reference of zero. (EX1005, 87; EX1003, ¶¶83, 102.)

Thus, MPEG-1's decoding a closed_gop flag or the Temporal Reference of zero for the first I-frame of the closed GOP and MPEG-1's disclosure of a closed GOP teaches or suggests 8[a]. (EX1003, ¶¶99-100.)

c. 8[b]

MPEG-1 teaches or suggests 8[b].

MPEG-1 starts the decoding of a video sequence from the first picture of a closed GOP (*first image frame of the independent sequence*). "If playback begins from a random point in the bitstream, the decoder should discard all the bits until it finds…a group of pictures start code." (EX1005, 120.) A closed GOP, like any

GOP, begins with a group of pictures start code. (EX1005, 43 (explaining that the GOP start code "identifies the beginning of a group of pictures").) Accordingly, when beginning playback from a random point in the bitstream, an MPEG-1 decoder should discard all bits until it finds the beginning of a closed GOP. A POSITA would have understood this to teach starting the decoding of the video sequence from the first picture of the closed GOP (*independent sequence*) because a closed GOP is "a series of one or more coded pictures." (EX1005, 32; EX1003, ¶103).

MPEG-1 also teaches *the video sequence is decoded without prediction from any image frame decoded prior to said first image frame*. Specifically, MPEG-1's closed GOPs are "decoded without using decoded pictures of the previous group for motion compensation." (EX1005, 86; EX1003, ¶104.)

MPEG-1 thus teaches or suggests starting decoding with the first picture of a closed GOP (*starting the decoding of the video sequence from the first image frame of an independent sequence*) and that a closed GOP is decoded without using decoded pictures of the previous group for motion compensation (*decoded without prediction from any image frame decoded prior to the first image frame*).[12]

---

[12] In a separate proceeding, PO argued that MPEG-1 does not teach 8[b] because the reference includes an example in which the decoder starts from an I-

d. 8[c]

MPEG-1 teaches or suggests 8[c].

Each picture in MPEG-1 has an identifier called a Temporal Reference that "is a ten-bit number which can be used to define the order in which the pictures must be displayed." (EX1005, 87, 44.) And MPEG-1 discloses or suggests that the Temporal Reference is decoded. For example, during forward playback, after decoding the group of pictures header, MPEG-1 says the decoder "will decode the first picture header in the group of pictures." (EX1005, 120.) And because the Temporal Reference is part of the picture header, the decoder decodes the Temporal Reference when it decodes the first picture header in the GOP. Furthermore, a POSITA would have understood that each picture in a GOP has a header (EX1005, 71) and that each of these headers would have been decoded as each picture was decoded. Consequently, a POSITA would have understood that each picture's Temporal Reference would be decoded. (EX1003, ¶105.)

---

picture in the middle of a video sequence. *Amazon.com,* IPR2024-00691, Paper 20, 45-47. This is irrelevant, however, because it ignores the exemplary closed GOP on which Petitioners rely. *See Bell Commc'ns Rsch.*, 55 F.3d at 622-23 (if the prior art sometimes, but not always, embodies a claim, it nonetheless teaches that claim).

MPEG-1 further discloses or suggests that Temporal Reference values are decoded according to a numbering scheme. During decoding, "slices and macroblocks in the picture are decoded and written into a display buffer" and then "displayed in the order defined by the temporal reference." (EX1005, 120.) A POSITA would have understood this to mean that the decoder outputs pictures in the order of their Temporal Reference. (EX1003, ¶106.) MPEG-1 also explains that pictures are assigned a Temporal Reference according to the following numbering scheme: "The first picture, in display order, in each [GOP] must have Temporal Reference equal to zero" and Temporal Reference is then "incremented by one for each picture in the group." (EX1005, 87, 44.) Because pictures are assigned a Temporal Reference according to a numbering scheme, and the decoder uses that numbering scheme to determine the order in which it should output the pictures, a POSITA would have understood the pictures and their corresponding Temporal Reference values (*identifier values*) are *decod[ed]…according to a numbering scheme*. (EX1003, ¶106.)

e. 8[d]

MPEG-1 teaches or suggests 8[d].

As explained in §VII.A.1.c, MPEG-1 discloses an exemplary closed GOP that begins with an I-frame in both decoding and display order. "For the earliest picture (in display order) in each group of pictures, the temporal-reference is reset to zero."

Nokia Exhibit 2012, p.35
ASUS v. Nokia
IPR2025-01279

(EX1005, 44, 87.) As shown in the figure below, the Temporal Reference is zero for the I-frame and then increments by one for each subsequent frame in display order.

Example (a) in display order	**I** 0	B 1	**P** 2	B 3	**P** 4
Example (a) in decoding order	**I** 0	**P** 2	B 1	**P** 4	B 3

MPEG-1, Fig.D.19(a)
("Examples of Groups of Pictures and Temporal References")

(EX1005, 87.)

This is true for every exemplary closed GOP in a bitstream. A POSITA would therefore have understood that when an MPEG-1 decoder decodes the next exemplary closed GOP, the Temporal Reference for the first picture in decoding order of that exemplary closed GOP would be reset to zero, as that picture would be the first picture in display order for that next GOP. (EX1003, ¶107.)

Thus, the Temporal Reference (*identifier value*) for the first picture in decoding order of every exemplary closed GOP (*the indicated first image frame of the independent sequence*) in a bitstream is reset to zero because that picture is also the first picture in the GOP in display order. (EX1003, ¶107.) Accordingly, MPEG-1 discloses *resetting the identifier value for the indicated first image frame of the independent sequence*.

In a separate proceeding, PO argued that MPEG-1 does not teach 8[d] for three reasons: (1) the encoder, not the decoder, resets the identifier; (2) only the identifier for a subsequent video sequence is reset; and (3) MPEG-1 does not always require that an I-picture (*the first image frame in decoding order*) have a Temporal Reference of zero. *Amazon.com*, IPR2024-00691, Paper 20, 50-53. All three arguments should be rejected.

Regarding argument (1), PO did not dispute that the Temporal Reference is reset for the first image frame of each independent sequence. *See id.*, 50-51. Nor did PO dispute that the encoder informs the decoder of the value of the Temporal Reference for each frame (including *the first image frame in decoding order*). *Id.* Nor is there any scenario described in MPEG-1 in which a compliant decoder would change the value of a syntax element set by a compliant encoder. Thus, if the decoder receives and decodes the value of a Temporal Reference for the first image frame and cannot change the value of that syntax element, it would have been obvious to a POSITA that an MPEG-1 decoder resets the Temporal Reference to zero when it decodes the first image frame in decoding order of the exemplary closed GOP. Otherwise, if a decoder unilaterally changed a Temporal Reference value, the decoder would not be able to correctly determine display order. (EX1003, ¶¶108-109.)

Regarding argument (2), Petitioners explain above that the Temporal Reference for every exemplary closed GOP in MPEG-1 is reset to zero. This is not only true for the Temporal Reference of the current video sequence, but also for every exemplary closed GOP. (EX1003, ¶¶108-109.)

Finally, argument (3) ignores the exemplary closed GOP on which Petitioners rely, which has an I-picture as the first image frame in decoding order, and the I-picture has a Temporal Reference of zero.

Thus, for the reasons discussed above, claim 8 is unpatentable as obvious over MPEG-1.

3. Claim 10

Claim 10 is substantively the same as claim 8, except it recites *a video decoder comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the video decoder to decode a compressed video sequence*.

a. 10[pre]

To the extent limiting, MPEG-1 teaches or suggests 10[pre]. (EX1003, ¶¶111-116.) As discussed in §VII.A.2.a for 8[pre], MPEG-1 teaches *decoding a*

Nokia Exhibit 2012, p.38
ASUS v. Nokia
IPR2025-01279

compressed video sequence. MPEG-1 also teaches the additional elements of 10[pre]. For example, MPEG-1 discloses multiple decoders:

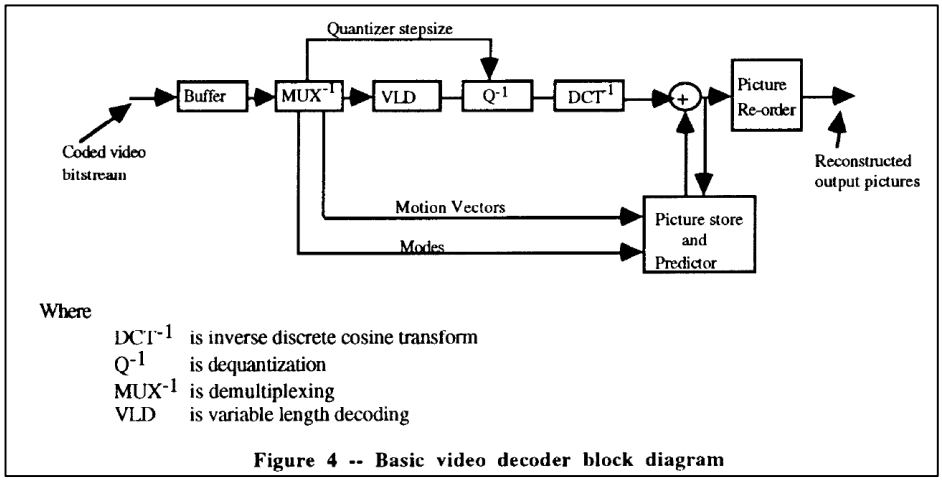


Figure 4 -- Basic video decoder block diagram

(EX1005, 16.)

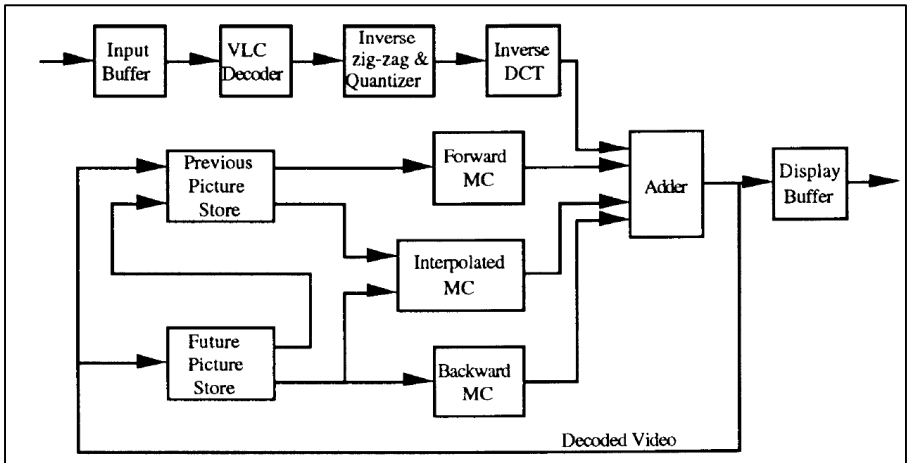


Figure D.7 -- Simplified decoder block diagram

(EX1005, 73.)

A POSITA would have found it obvious to implement MPEG-1's decoder as *a system comprising at least one processor and at least one memory including computer program code configured to, with the at least one processor, cause the*

video decoder to decode a compressed video sequence based on a POSITA's background knowledge at the relevant time. (EX1003, ¶112.) As part of that background knowledge, a POSITA would have been aware of decoders with the recited hardware that existed before the priority date of the '321 patent.

One example of such a decoder was C-Cube's CL480PC MPEG-1 audio/video decoder. (EX1007, 1.) The CL480PC decoded video using a combination of processors. (EX1007, 3.) One of the processors used code stored in a DRAM that was loaded into on-chip memory. (EX1007, 3.) Given that C-Cube's CL480PC was an operating MPEG-1 decoder, it supports that a POSITA would have understood MPEG-1's decoder to be implemented with a combination of hardware and software, including a processor, a memory, and computer code. (EX1003, ¶113.) Indeed, this configuration featuring one or more processors implementing code stored in a DRAM for the decoding process was a common and obvious configuration for an MPEG-1 decoder. (EX1003, ¶¶113.)

MPEG-1 also suggests implementing its decoder using the processor, memory, and computer program code of a personal computer. (EX1005, 12.) A POSITA would have recognized that MPEG-1 suggests decoding compressed video using a processor in personal running computer code stored in memory. (EX1003, ¶¶114-16; EX1008, 6:56-65, 7:53-58; EX1009, 60; EX1010, 3.)

Accordingly, it would have been obvious to a POSITA to implement MPEG-1's disclosed decoders as a *video decoder comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the video decoder to decode a compressed video sequence*.

b. 10[a]

This limitation is substantively identical to 8[a], and MPEG-1 teaches it for the same reasons. §VII.A.2.b.

c. 10[b]

This limitation is substantively identical to 8[b], and MPEG-1 teaches it for the same reasons. §VII.A.2.c.

d. 10[c]

This limitation is substantively identical to 8[c], and MPEG-1 teaches it for the same reasons. §VII.A.2.d.

e. 10[d]

This limitation is substantively identical to 8[d], and MPEG-1 teaches it for the same reasons. §VII.A.2.e.

For the reasons discussed above, claim 10 would have been obvious based on MPEG-1.

Nokia Exhibit 2012, p.41
ASUS v. Nokia
IPR2025-01279

4. Claim 11

Claim 11 is substantively the same as claim 8, except it recites *a computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence*.

a. 11[pre]

To the extent limiting, MPEG-1 teaches or suggests 11[pre]. As discussed in §VII.A.2.a for 8[pre], MPEG-1 teaches *decoding a compressed video sequence*. MPEG-1 also teaches the additional elements of 11[pre]. For example, MPEG-1 discloses multiple decoders:

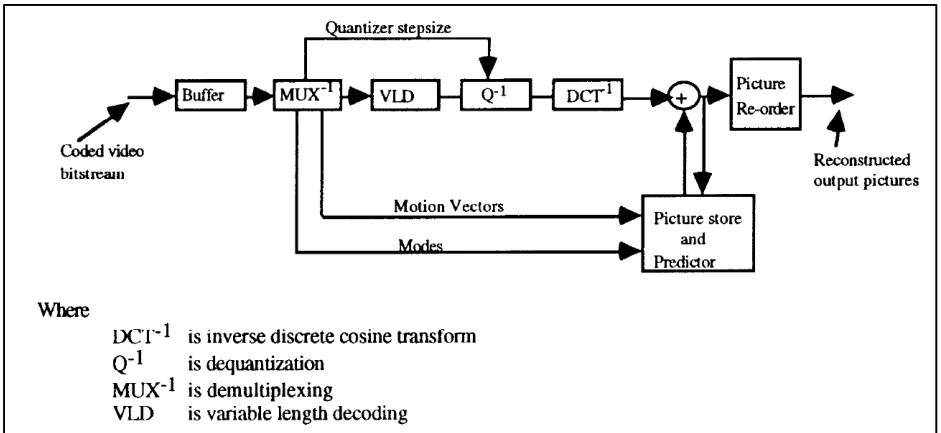


Figure 4 -- Basic video decoder block diagram

(EX1005, 16.)

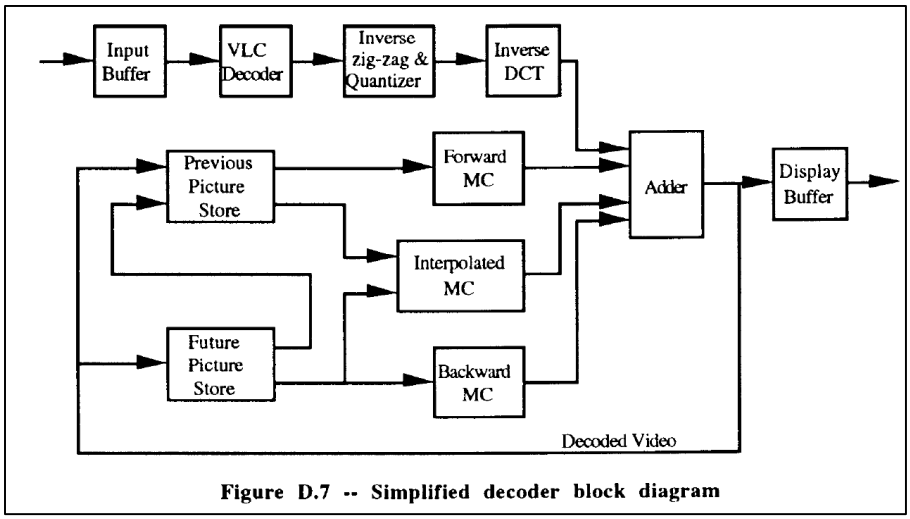


Figure D.7 -- Simplified decoder block diagram

(EX1005, 73.)

A POSITA would have found it obvious to implement the disclosed video decoder in MPEG-1 as a computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence. (EX1003, ¶117.) As explained in §VII.A.3.a, multiple examples of MPEG-1 decoders were well known in the art before the priority date of the '321 patent. For example, C-Cube's CL480PC MPEG-1 audio/video decoder decoded video using a combination of processors. (EX1007, 3.) One of the CL480PC's processors used microcode stored in a DRAM that was loaded into on-chip memory. (EX1007, 3.) A POSITA would have understood that this microcode was *a computer program product…executable in a data processing device, for decoding a compressed video sequence* and that the DRAM on which it was stored was *a non-transitory computer readable medium*. (EX1003, ¶¶117-18.)

Moreover, a POSITA would have understood the configuration of C-Cube's CL480PC to be a common and obvious configuration existing in the art before the priority date of the '321 patent. (EX1003, ¶119.)

As discussed above for 10[pre], MPEG-1 also suggests implementing its decoder as a *computer program product* using the processor, memory, and computer program code of a personal computer. (EX1005, 12; EX1003, ¶¶119-21; EX1008, 6:59-65, 7:53-58; EX1009, 60; EX1010, 3.)

Accordingly, a POSITA would have found it obvious and would have been motivated to implement any of MPEG-1's disclosed decoders as *a computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence*. (EX1003, ¶¶117-21.)

b. 11[a]

This limitation is substantively identical to 8[a], except it recites *a computer program code for* decoding an indication rather than the act of decoding an indication. As explained in §VII.A.2.b, MPEG-1 teaches *decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence*. And as explained in §VII.A.5.a, a

POSITA would have recognized that the decoding would be carried out by *a computer program product*, which would include *computer program code* such as microcode or C-code. MPEG-1 therefore teaches or suggests 11[a].

c. 11[b]

This limitation is substantively identical to 8[b], except it recites *a computer program code for* starting the decoding rather than the act of starting the decoding. As explained in §VII.A.2.c, MPEG-1 teaches *starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame*. And as explained in §VII.A.2.a, a POSITA would have recognized that the decoding process would be carried out by *a computer program product*, which would include *computer program code* such as microcode or C-code. MPEG-1 therefore teaches or suggests 11[b].

d. 11[c]

This limitation is substantively identical to 8[c], except it recites *a computer code for* decoding identifier values rather than the act of decoding identifier values. As explained in §VII.A.2.d, MPEG-1 teaches *decoding identifier values for image frames according to a numbering scheme*. And as explained in §VII.A.2.a, a POSITA would have recognized that the decoding process would be carried out by

Nokia Exhibit 2012, p.45
ASUS v. Nokia
IPR2025-01279

a computer program product, which would include *computer program code* such as microcode or C-code. MPEG-1 therefore teaches or suggests 11[c].

e. 11[d]

This limitation is substantively identical to 8[d], except it recites *a computer program code for* resetting the identifier value rather than the act of resetting the identifier value. As explained in §VII.A.2.e, MPEG-1 teaches *resetting the identifier value for the indicated first image frame of the independent sequence*. And as explained in §VII.A.2.a, a POSITA would have recognized that the decoding process would be carried out by *a computer program product*, which would include *computer program code* such as microcode or C-code. MPEG-1, therefore, teaches or suggests 11[d].

For the reasons discussed above, claim 11 would have been obvious based on MPEG-1. (EX1003, ¶¶117-21.)

B. Ground 2: Kim Renders Obvious Claims 8, 10, and 11

1. Kim

Kim discloses "a device and method for recording and reproducing video data on and from an optical disk using a Time Lapse recording/reproduction process." (EX1006, 1:10-12.) Relevant here, Kim describes a "reproducing method" that involves decoding a bit stream of compressed video data in response to a user's request and outputting an analog image signal. (EX1006, 5:8-17, 5:29-34, 5:55-6:8.)

Nokia Exhibit 2012, p.46
ASUS v. Nokia
IPR2025-01279

To support its reproducing method, Kim describes a bitstream with pictures grouped into a "logical structure" based on dependencies, illustrated in Figure 5A:

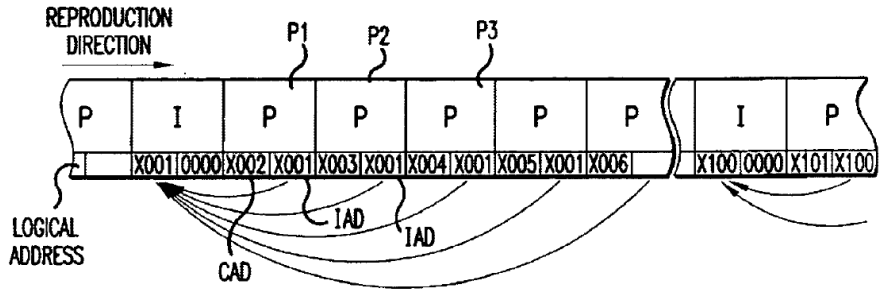


FIG.5A

(EX1006, Fig. 5A.) The logical structure comprises an I-picture followed by P-pictures that depend on it. (EX1006, Fig. 5A, 4:46-52.) Kim's I- and P-pictures have similar characteristics to MPEG-1's I- and P-pictures discussed above. (*See* EX1006, 4:15-19, 6:55-58; EX1003, ¶¶122-23.) Specifically, in Kim, each I-picture is an "intra-picture" and therefore "carries the entire image information of a frame." (EX1006, 3:60-61.) Kim's I-pictures, therefore, do not depend on other pictures to be decoded and output as an analog image signal. (EX1003, ¶123.) In addition, in Kim, each P-picture is a "predictive-picture" (EX1006, 2:22) encoded based on "the moving vector and the position value of the corresponding (or previous) I-frame" (EX1006, 4:28-33). Kim's P-pictures, therefore, require the data from their corresponding I-pictures moving vectors to be restored as full screen images. (EX1006, 5:55-59.)

Kim further discloses that the bitstream is comprised of data structures that associate each picture with address fields used to identify the picture and its corresponding reference picture. Kim depicts these data structures in Figure 6:

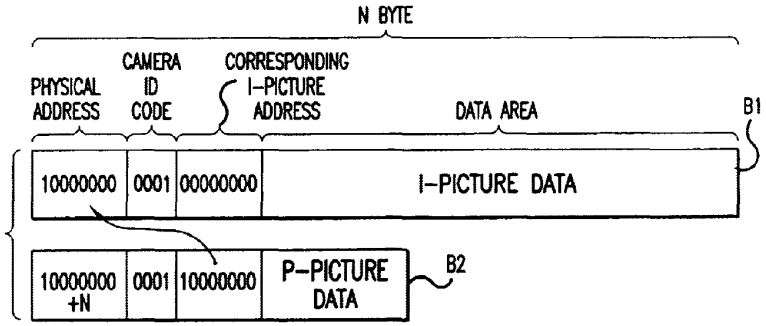


FIG.6

(EX1006, Fig. 6, 4:59-61.)

Kim explains that each P-picture has a current address ("CAD") that indicates the current address of the P-picture, and an I-picture reference address ("IAD") that indicates the address of the corresponding (or previous) I-frame. (EX1006, 4:46-52.) I-pictures also have a CAD indicating the picture's current address (EX1006, 4:65-5:1, Figs. 5A, 6 ("physical address")), but the I-picture's IAD is zero (EX1006, Figs. 5A (showing "0000"), 6 (showing "00000000" for the "corresponding I-Picture address")). Figure 5A below is annotated to highlight the IAD values in red and CAD values in green.

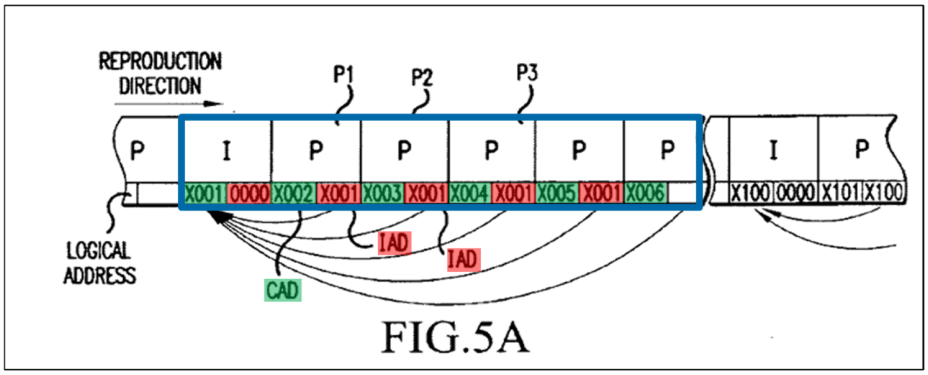


FIG.5A

(EX1006, Fig. 5A.)

As shown below in Kim's Figure 4, Kim sets forth two paths for decoding the disclosed bitstream depending on whether the user requests image data that corresponds to an I-picture or a P-picture. (EX1003, ¶¶126.) When the request corresponds to an I-picture, the picture is decoded and output as an image signal. (EX1006, 5:66-6:3, Fig. 4 ("no" path annotated in red).) When the request is directed to a P-picture, the system "detects the address of the corresponding I-picture frame…and reproduces based on the detected address [of that] I-picture." (EX1006, 5:38-48; Fig. 4 ("yes" path annotated in blue).)

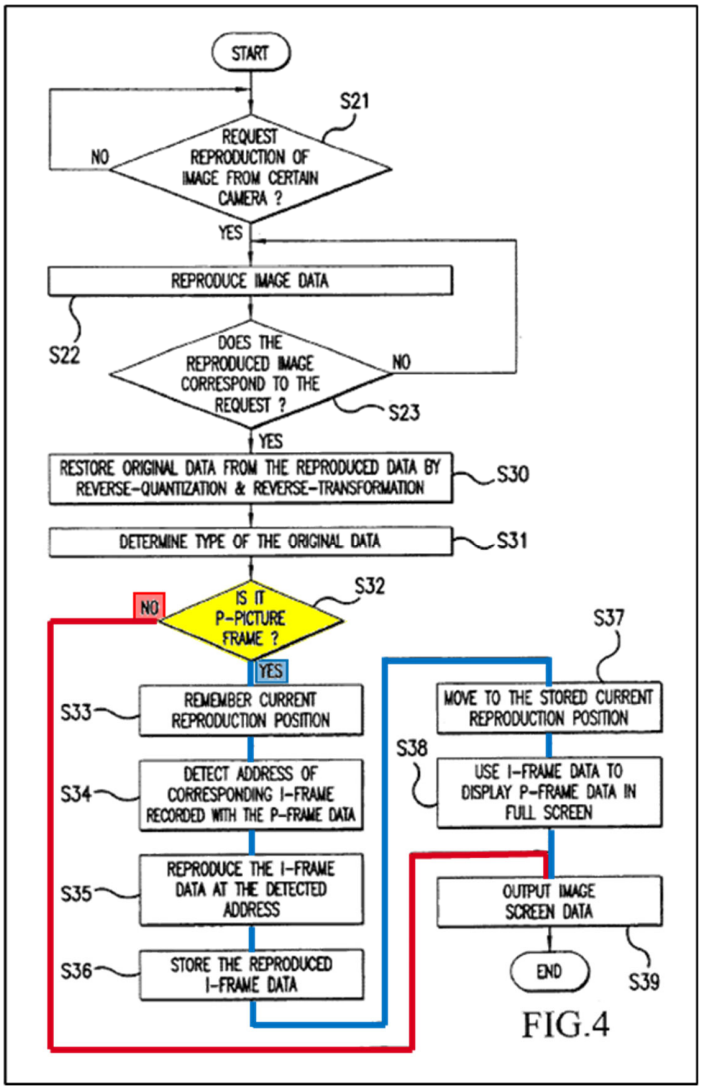


FIG.4

(EX1006, Fig.4.)

2. Claim 8

a. 8[pre]

To the extent limiting, Kim discloses or suggests 8[pre]. (EX1003, ¶¶127-28.) Kim is directed to "reproducing video data on and from an optical disk using a Time Lapse recording/reproduction process." (EX1006, 1:11-12.) Kim "provide[s] a recording method for compressing and intermittently recording image data on a

recording medium with a maximum recording capacity." (EX1006, 2:3-6.) Kim further "provide[s] a reproduction method for reproducing [the] compressed intermittent data from a recording medium." (EX1006, 2:7-9.) A step in the reproduction method includes decoding image data. (EX1006, 5:29-31.) Accordingly, Kim discloses or suggests a *method for decoding* its compressed intermittent data (*a compressed video sequence*).

b. 8[a]

Kim teaches or suggests 8[a]. (EX1003, ¶¶129-31.) Kim's Figure 5A (reproduced below) shows an example of a logical data structure of a digital data stream. (EX1006, 2:48-51.) As explained below, this figure shows the recited *indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence*.

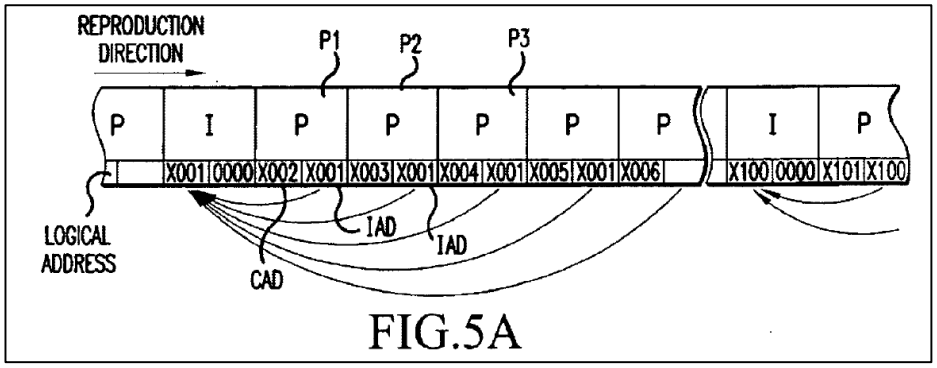


FIG.5A

(EX1006, Fig. 5A.)

Kim's bitstream disclosed includes I-pictures and P-pictures. (EX1006, Fig. 5A, 2:17-27.) I-pictures are coded based only on information within themselves.

(EX1006, 3:60-61 ("The I-picture frame (intra-frame) data carries the entire image information of a frame.").) P-pictures (also referred to in Kim as "P-frames") are similar to those in MPEG-1 and are coded with "the moving vector and the position value of the corresponding (or previous) I-frame." (EX1006, 4:15-27.) Kim's grouping of an I-picture followed by P-pictures that refer to it correspond to *an independent sequence*. (EX1003, ¶¶133-34.)

In Kim's *independent sequence*, each P-picture following an I-picture depends on that I-picture for motion prediction. (EX1006, Fig. 5A, 4:46-52; EX1003, ¶131.) Kim explains that "each P-picture frame…has a current address CAD indicating the current address of the P-picture, and an I-picture reference address IAD indicating the address of the corresponding (or previous) I-frame." (EX1006, 4:46-52.) Thus, each P-picture's IAD identifies the I-picture on which the P-picture depends for motion compensation. (EX1003, ¶131.) This is shown in Figure 5A, where every P-picture's IAD identifies the address of the preceding I-picture and shows these dependencies as arrows pointing from each P-picture back to the preceding I-picture. (EX1006, Fig. 5A (IAD highlighted in red; CAD highlighted in green; independent sequence outlined by blue box); EX1003, ¶131.)

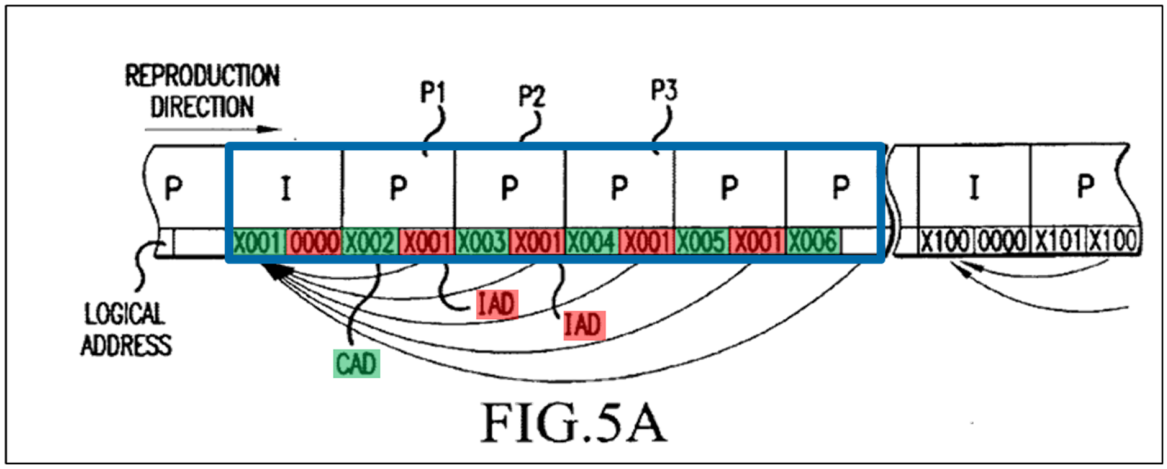


FIG.5A

Because every P-picture following the I-picture depends on that I-picture for motion prediction in the sequence shown in Figure 5A, all motion-compensated temporal prediction references of the sequence shown in Figure 5A refer only to image frames within the sequence. (EX1003, ¶132.) Accordingly, Kim discloses or suggests a group of frames where every P-picture depends only on the preceding I-picture for motion prediction (*an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within the independent sequence*).

Kim also teaches the recited *indication*. (EX1003, ¶133.) Specifically, as shown in Figure 5A, an IAD of 0000 or zero is *an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence*. Kim discloses that each image frame has a CAD and an IAD. (EX1006 4:46-52, 4:59-5:7, Figs. 5A, 6.) Kim's Figure 5A shows that each I-picture has an IAD of 0000. (EX1006, Fig. 5A; EX1006, 4:59-5:1 (explaining that for I-pictures

"the address of the corresponding I-picture frame [(*i.e.*, the IAD)]…is *zero* or absent").) The IAD for each P-picture, on the other hand, is the CAD (current address) of the preceding I-picture. (EX1006, 4:48-52.) Because each I-picture starts an independent sequence, the IAD of 0000 or zero indicates that a given I-picture starts an independent sequence (is *the first image frame, in decoding order, of an independent sequence*). Figure 5A shows the CAD for the I-picture and IAD for the P-pictures of the first independent sequence highlighted in yellow and shows the CAD for the I-picture and IAD for the P-pictures of the second independent sequence highlighted in light blue.

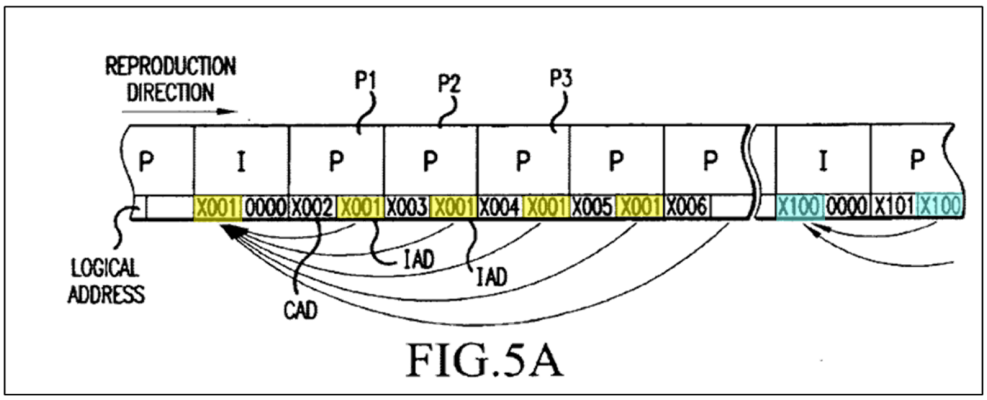


FIG.5A

(EX1006, Fig. 5A.)

A POSITA would have understood that Kim's decoder decodes the IAD of the image frames. (EX1003, ¶134.) Figure 5A provides an "example[] of a logical data structure of a digital data stream," and "FIG. 6 shows an example of a data structure of the bit stream shown in FIG[]. 5A" (EX1006, 2:48-51, 4:59-61). Kim then explains the process for decoding this bitstream. (EX1006, 5:8-6:8, Fig. 4.) A

POSITA would have understood that decoding the bitstream in Figure 5A would include decoding the components of that bitstream, including the IAD values. (EX1003, ¶¶132-33.) Accordingly, Kim teaches or suggests 8[a] by decoding the IAD value of 0000 for the I-picture in Kim's Figure 5A.

Taken together, Kim's disclosures of decoding the IAD value of 0000 for the I-picture and grouping of I-pictures followed by P-pictures that depend on the corresponding I-picture for that group teach or suggest *decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence*.

In a separate proceeding, PO argued that Kim does not teach 8[a] for three reasons: (1) if a P-picture (not an I-picture) is selected first, it does not have an IAD of 0000 but is the first image frame in decoding order; (2) if an I-picture is selected first, the IAD is not decoded; and (3) Kim discloses searching an optical medium using a physical address rather than the logical address that includes the IAD. *Amazon.com*, IPR2024-00691, Paper 20, 56-59. All three arguments fail.

PO's argument (1) appears to interpret *decoding order* based on where the decoder **starts** decoding frames—which may not go to completion due to reference frame dependency—rather than where the decoder completely **decodes** image

frames. *See id.*, 56-57. Such an interpretation finds no support in the '321 patent and departs from the plain-and-ordinary meaning of "decoding order" (or the synonym, "bitstream order") to a POSITA: "the order which the decoder requires them to decode the bitstream." (EX1005, 71; EX1003, ¶135.) As explained above, Kim's I-picture is decoded before the P-picture that depends on it, even if decoding the P-picture starts but cannot successfully finish. A POSITA would have understood that the I-picture is the first image frame in decoding order, even when a P-picture is selected (*id.*), and in the process flow for decoding a P-picture, the decoder decodes the IAD value 0000 (*indication*) when it decodes the I-picture on which the P-picture depends. (EX1003, ¶¶135-36.)

As to argument (2), PO misapprehended Kim's teachings. According to PO, only after Kim's decoder reproduces the requested image data does it look at what "type" the original data is, at which point the decoder would not need to determine the type of data (I- or P-) to begin decoding. But Kim explains (with reference to Figure 4 below) that the decoder does not know until step S30—where image data is transmitted to the decoder—whether the frame is an I-picture or P-picture.

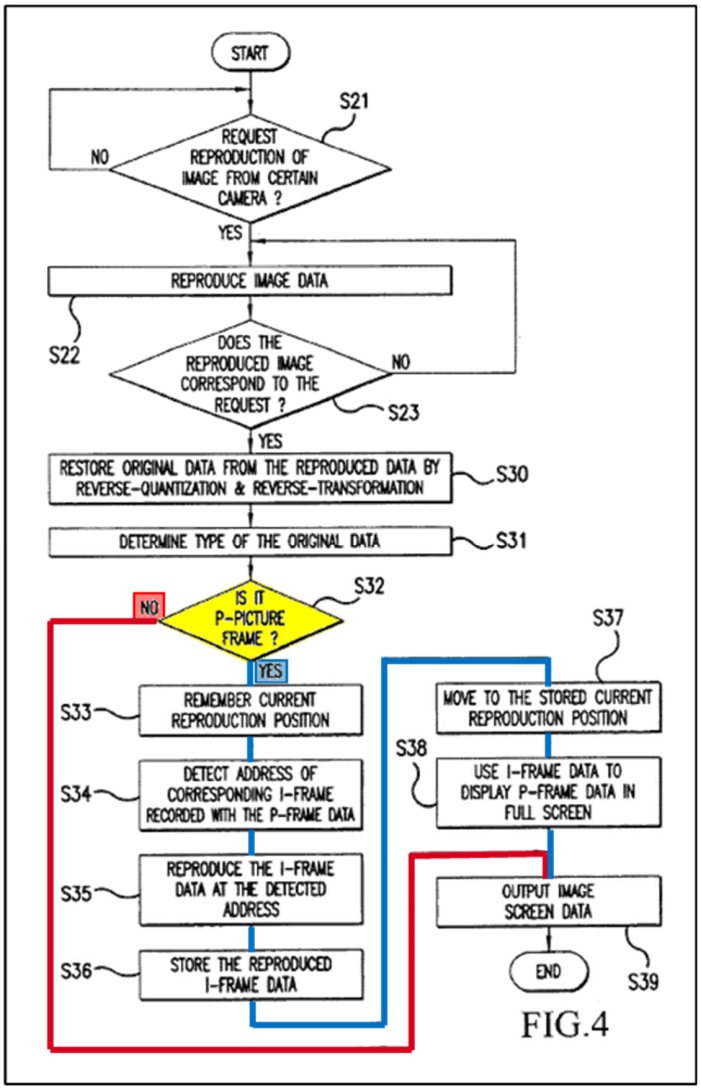


FIG.4

(EX1006, Fig.4.) If the image data identifies an I-picture (*first image frame in decoding order*), the IAD of 0000 is decoded. And as explained above, if a P-picture is selected, the IAD of the I-picture it depends on is still decoded; otherwise, the selected P-picture could not be fully decoded. (EX1003, ¶¶135-36.)

Finally, argument (3) ignores that Kim must decode the logical address (e.g, IAD) before it can populate the data structure at a physical address. (EX1003, ¶137.)

Accordingly, Kim teaches a POSITA that the IAD/CAD is decoded to populate the data structure illustrated in Figure 6.

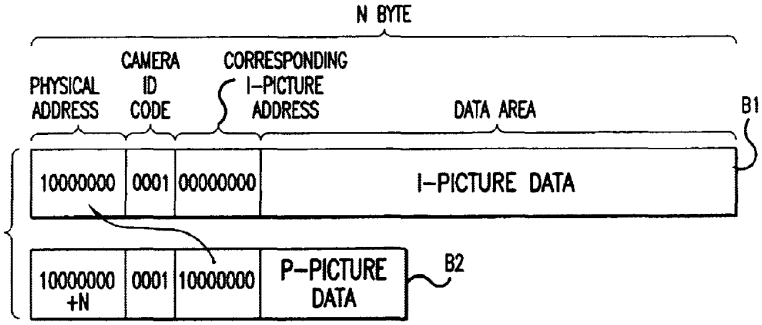


FIG.6

(EX1006, Fig.6.)

c. 8[b]

Kim discloses or suggests 8[b]. (EX1003, ¶¶138-48.)

As discussed in §VII.B.2.b for 8[a], Kim's disclosure of a grouping of an I-picture followed by P-pictures teaches *an independent sequence*. As to 8[b], Kim teaches starting decoding of that grouping with the I-picture (*starting the decoding of the video sequence from said first image frame of the independent sequence*). Specifically, as shown in Figure 4, Kim discloses two paths for decoding an image in response to a user requesting playback. Either path discloses or suggests 8[b].

A POSITA would have understood that Kim's Figure 4 process is applied—or would have found it obvious to apply Figure 4—to each image frame in a video sequence, including the sequence in Kim's Figure 5A. Specifically, a POSITA would have been motivated to apply Kim's Figure 4 process to Figure 5A video data

rather than a single image so that the Figure 4 process could be applied to more types of input data. (EX1003, ¶140.) For example, the flowchart in Figure 4 depicts a reproducing method for a data structure, such as shown in Figure 6. (EX1006, 6:8-10.) And Kim further discloses that Figure 6 shows an example of a data structure of the bitstream shown in Figure 5A. (EX1006, 4:59-60.) A POSITA would have understood that the decoding steps in Figure 4 could be repeated for each image in an independent sequence like Figure 5A. *See Perfect Web Techs., Inc. v. InfoUSA, Inc.*, 587 F.3d 1324, 1330 (Fed. Cir. 2009) (POSITA's common sense taught repeating an element existing in the prior art until success was achieved); *see also B/E Aerospace, Inc. v. C&D Zodiac, Inc.*, 962 F.3d 1373, 1380-81 (Fed. Cir. 2020) (same). Moreover, as explained for Ground 1, decoding video sequences including closed GOPs was well-known at the relevant time. (EX1003, ¶140.) A POSITA would have found this combination to be no more than applying prior-art elements (Kim's Figure 4 process to the Figure 5A data structure) according to their known functions to achieve predictable results. (EX1003, ¶140.)

Kim's Figure 4 is reproduced below:

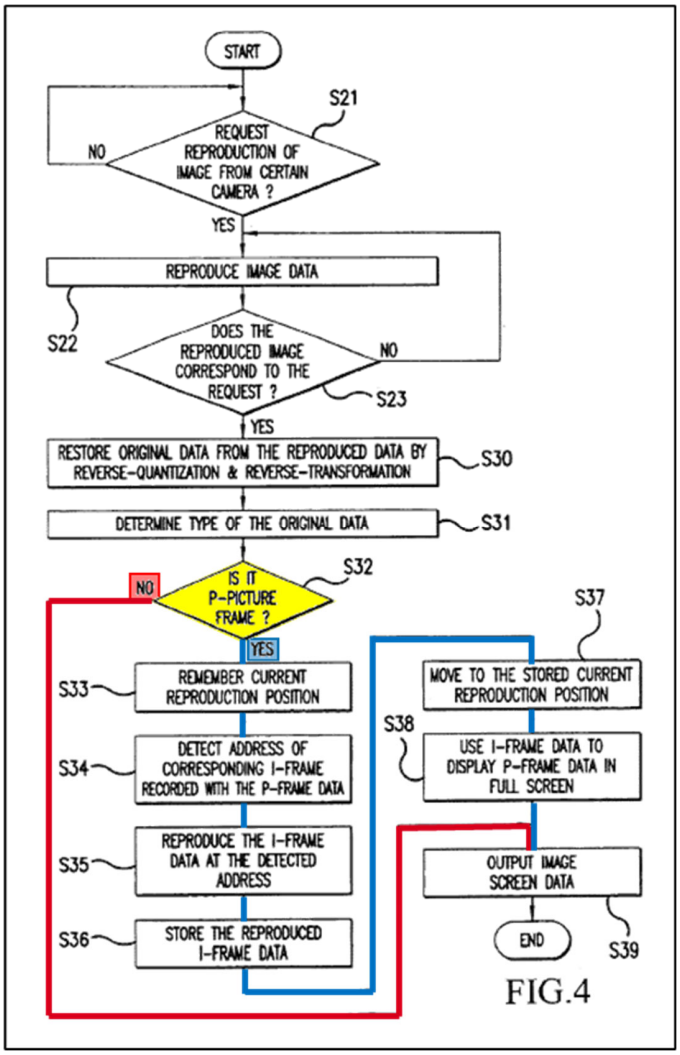


FIG.4

(EX1006, Fig. 4.)

Kim's first path (red) discloses that the first image decoded and output for display is an I-picture. Kim explains that "in Step S32, if the detected image data is not a P-picture frame, then the detected image data is decoded by the decoder 111 as described above, and output as an analog image signal from the D/A converter 91 in Step S39." (EX1006, 5:66-6:3.) Kim's use of the phrase "not a P-picture" means that the picture is an I-picture because Kim's sequence includes only I- and P-

pictures. (EX1006, Figs. 5A, 6.) As explained above in §VII.B.2.b, Figure 5A shows that each independent sequence begins with an I-picture. Accordingly, this path, when applied to Figure 5A's independent sequence (which begins by decoding an I-picture), discloses *starting the decoding of the video sequence from said first image frame of the independent sequence*. (EX1003, ¶¶141-42.)

Although Kim's disclosure of the first path is sufficient to teach or suggest this limitation, the second path (blue) also discloses *starting the decoding of the video sequence from said first image frame of the independent sequence*. The second path reflects the system's behavior when the user requests reproduction of a P-picture. In the second path, "if the currently restored image data is detected to be P-picture frame data in Step S32, the controller 150 stores the current reproduction position in Step S33." (EX1006, 5:38-41.) The system then detects the address of the corresponding I-picture on which the P-picture depends for motion compensation. (EX1006, 5:41-48.) The system decodes that I-picture, after which the P-picture is restored as a full-screen image using the I-picture frame data and moving vector and is ultimately output as a reproduction signal. (EX1006, 5:49-65.) A POSITA would have understood this second path when applied to Figure 5A to disclose *starting the decoding of the video sequence from said first image frame of the independent sequence* because the system decodes the I-picture (*the first image frame of the independent sequence*) before it can produce the P-picture that depends

on it. (EX1003, ¶¶143-44.) Thus, based on either path from Step S32 in Kim's Figure 4 when applied to Figure 5A's independent sequence, Kim discloses or suggests *starting the decoding of the video sequence from said first image frame of the independent sequence*.

As discussed above, a POSITA would have found it obvious to apply Kim's Figure 4 process to each image frame in a video sequence, including the sequence in Figure 5A. Alternatively, Figure 4 itself contemplates a video sequence that includes an I-picture and a P-picture because, at Step 32, if the user requests reproduction of a P-picture, the system detects the address of the corresponding I-picture on which the P-picture depends for motion compensation and decodes that I-picture. (EX1006, Fig. 4 (showing "yes" branch at Step S32), 5:38-52.) Thus, even without applying Figure 4's process to all the pictures of an independent sequence in Figure 5A, a POSITA would have understood that Figure 4's process applies to an independent sequence consisting of a P-picture and a corresponding I-picture. (EX1003, ¶144.) This is sufficient to teach *starting the decoding of the video sequence from said first image frame of the independent sequence*.

Kim also discloses or suggests that *the video sequence is decoded without prediction from any image frame decoded prior to said first image frame*. As explained in §VII.B.2.b, each independent sequence in Kim's bitstream consists of an I-picture followed by a series of P-pictures that depend on the I-picture for

prediction. (EX1006, Fig. 5A.) Because every P-picture refers to the first I-picture in each independent sequence, no P-picture requires prediction from any image frame decoded prior to the first I-picture. (EX1003, ¶145.)

In sum, either decoding path described in Kim's Figure 4 when applied to Figure 5A's independent sequence, or alternatively the "yes" path from Step 32 in Kim's Figure 4 when applied to an independent sequence with a P-picture and corresponding I-picture, discloses *starting the decoding of the video sequence from said first image frame of the independent sequence*. And either path discloses decoding the independent sequence *without prediction from any image frame decoded prior to said first image frame*. (EX1003, ¶¶138-46.)

In a separate proceeding, PO argued that Kim does not disclose 8[b] because the decoder starts decoding the P-picture before it decodes the I-picture on which it depends. *Amazon.com*, IPR2024-00691, Paper 20, 60-61. But as explained above for 8[a], a POSITA reading Kim would understand that an I-picture is the first frame in decoding order, even if the user selects a P-picture. (EX1003, ¶147.) In addition, PO argued that Kim does not meet 8[b] because: (1) if the user selects an I-picture, no sequence is decoded; and (2) if Kim only decoded I-pictures, the IAD would not be unique. *Amazon.com*, IPR2024-00691, Paper 20, 62. Both of these arguments misunderstand Petitioners' obviousness ground, which does not rely upon a decoder

from Kim that only decodes I-pictures. For example, Figure 5A shows a sequence of one I-picture followed by P-pictures.

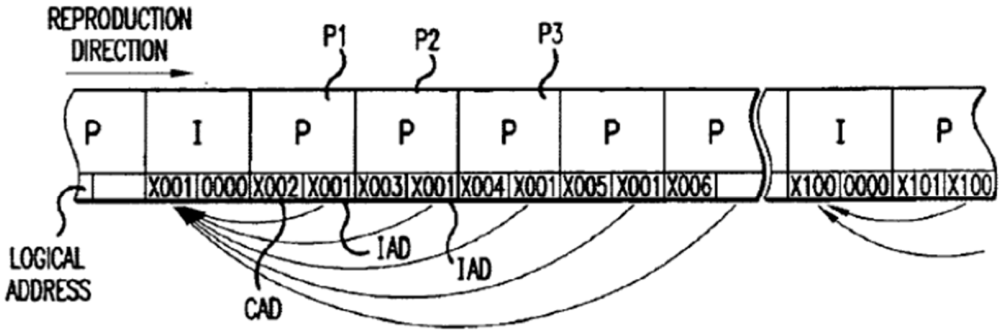


FIG.5A

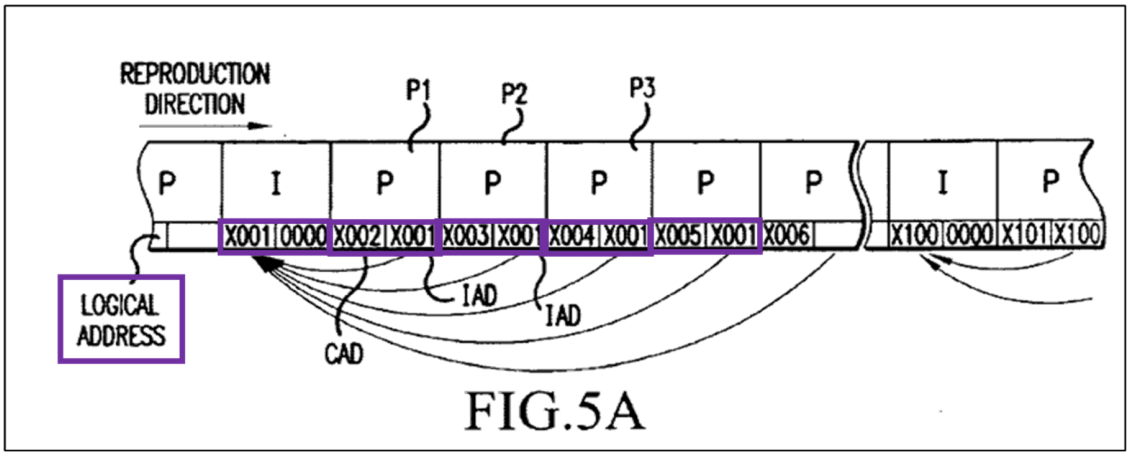
(EX1006, Fig. 5A.) As explained above, if a user selected the I-picture, followed by P-pictures, Kim's decoder would start decoding the sequence in Figure 5A with the first image frame (the I-picture), just as 8[b] requires. (EX1003, ¶148.)

d. 8[c]

Kim teaches or suggests 8[c]. (EX1003, ¶¶149-52.) Specifically, Kim's "logical addresses" are *identifier values for image frames*, and they are decoded *according to a numbering scheme*.

Kim teaches or suggests that pictures have "logical addresses" assigned according to a numbering scheme. As shown in Figure 5A, each picture has a "logical address" comprised of the picture's IAD and CAD. (EX1006, Fig.5A (annotated below with logical addresses outlined in purple).) For I-pictures and P-pictures, the CAD component of the logical address provides the current address of

a picture. (EX1006, 4:48-52.) CAD increments by one for each picture in reproduction order. (EX1006, Fig. 5A (showing that the first I-picture has a CAD of 0001, that the following P-picture has a CAD of 0002, and that CAD continues to increment by 1 for each subsequent P-picture in the group).) For I-pictures, the IAD component is zero. (*Id.*) And for P-pictures, the IAD provides the CAD of the corresponding I-picture. (EX1006, 4:48-52.) Accordingly, the pictures (*image frames*) in Kim's disclosed bitstream have logical addresses that follow *a numbering scheme*. (EX1003, ¶150.)



(EX1006, Fig. 5A.)

Kim discloses or suggests that logical addresses are *decod[ed]…according to a numbering scheme*. (EX1003, ¶151.) Specifically, when a P-picture is decoded, "the decoder detects the address of the corresponding I-picture frame…and reproduces based on the detected address." (EX1006, 5:41-46.) Thus, the decoder uses the logical addresses to determine the order in which pictures are decoded and

reproduced, and in reproducing the pictures in this order, the decoder also decodes the logical addresses. (EX1003, ¶151.)

In a separate proceeding, PO argued that Kim does not teach 8[c] because it decodes physical addresses—which, according to PO, "are not according to a numbering scheme"—rather than logical ones. *Amazon.com*, IPR2024-00691, Paper 20, 63-65. This argument is based on an "optical disc" example from Kim. *Id.* (citing EX1006, 5:41-44). PO's arguments are wrong for two reasons: first, Petitioners rely upon Kim's logical addresses as the claimed *identifier values*, so it is irrelevant whether Kim's physical addresses follow *a numbering scheme*. Second, PO's citation to a single example in Kim does not refute Petitioners' obviousness argument, particularly where, as here, a POSITA would understand that Kim must decode the "logical address" to identify the corresponding physical address. (EX1003, ¶152.)

e. 8[d]

Kim teaches or suggests 8[d]. (EX1003, ¶¶153-58.) Specifically, Figure 5A shows that the IAD component of each logical address is reset to zero for each I-picture. (EX1006, Fig. 5A (annotated below to show IAD component of each I-picture's logical address in red and the whole logical address outlined in purple).) Because each I-picture is the first image frame of an independent sequence, Kim discloses or suggests that the IAD component of each logical address (*identifier*

value) is reset for the indicated first image frame of the independent sequence. (EX1003, ¶153.)

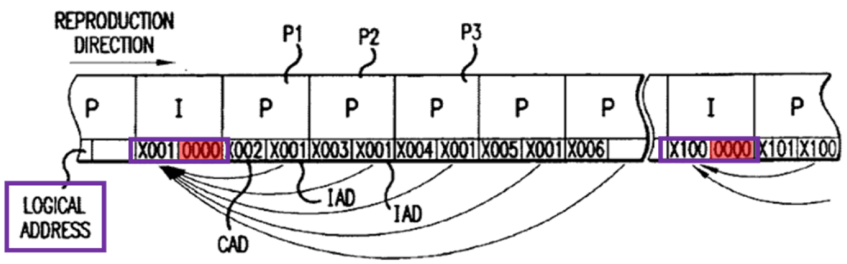


FIG.5A

(EX1006, Fig.5A.)

To the extent Patent Owner argues that Kim's disclosure of resetting a component of an identifier value does not teach or suggest this limitation, that is incorrect because resetting a component of the identifier value of an indicated first image frame is consistent with the '321 patent disclosure. (EX1003, ¶¶154-56.) In fact, the only two disclosures in the '321 patent addressing more than one independent sequence, Figures 2 and 3a, show only a component of an identifier value being reset.

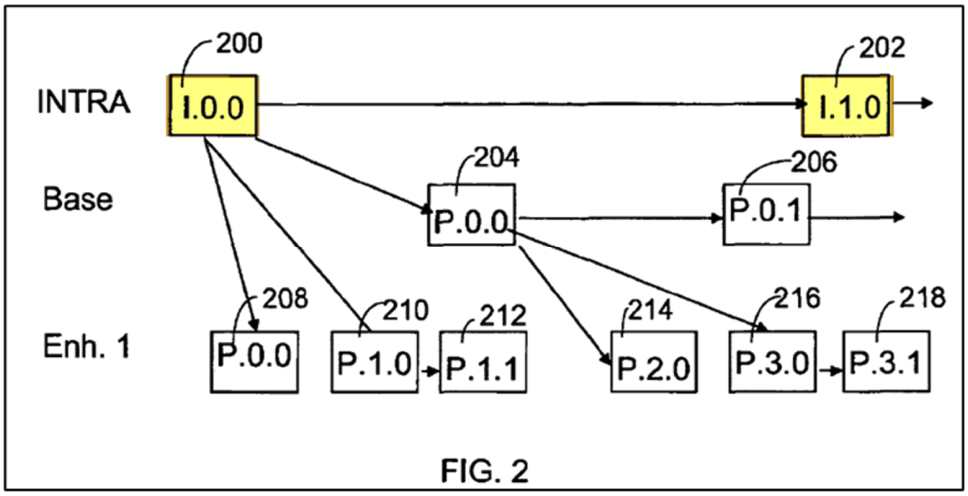


FIG. 2

(EX1001, Fig.2.) As shown, I-frames 200 and 202 reflect the first image frames of independent sequences. (EX1001, Fig.2.) I-frame 200 has a type I and a subsequence identifier and image number of (0.0), and I-frame 202 has a type I and subsequence identifier and image number of 1.0. (EX1001, 8:22-40.) The ’321 patent explains that the identifier for I-frame 202 is therefore I.1.0—a combination of the type, subsequence identifier, and image number—and the next I-frame would have an identifier of I.2.0. (EX1001, 8:41-43.) Thus, only a component of the ’321 patent’s disclosed identifiers, the image number, resets.

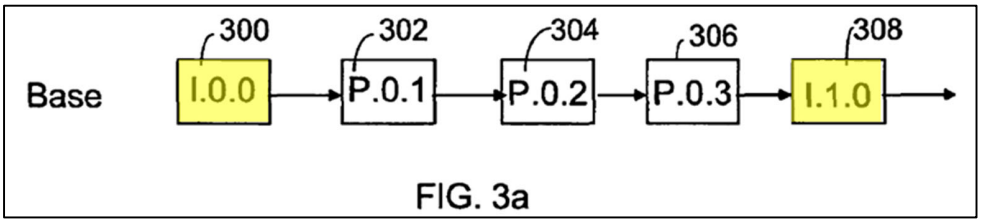


FIG. 3a

Figure 3a of the ’321 patent likewise shows that, within the meaning of the ’321 patent, *resetting an identifier value* includes resetting a component of an

identifier value. (EX1003, ¶156.) The '321 patent states that "[t]he video sequence comprises a first image frame 300 which is an I-frame (I.0.0) and which thus initiates a first sub-sequence." (EX1001, 12:24-26.) Furthermore, "a second subsequence" is initiated by "an I-frame (I.1.0) coded therein, i.e. an I-frame 308." (EX1001, 12:31-34.) Thus, again, only a component of the identifier value is reset when a new subsequence is initiated, confirming that Kim teaches *resetting an identifier value* in the same way as the '321 patent discloses.

Accordingly, Kim teaches or suggests *resetting the identifier value for the indicated first image frame of the independent sequence* through its disclosure that the IAD portion of a picture's logical address is reset for the first frame of an independent sequence.

In a separate proceeding, PO argued that Kim does not teach or suggest 8[d] because (1) Kim only discloses that its encoder resets the identifier value; and (2) Kim does not teach that its reproducing method resets the IAD. *Amazon.com*, IPR2024-00691, Paper 20, 65-68. Both of these arguments should be rejected.

While the MUX of Kim's encoder outputs the address of I- and P-pictures in the bitstream (EX1006, 4:38-42), that does not preclude Kim's decoder from resetting the IAD for each I-picture (*first image frame*). As explained above, Kim teaches that the bitstream transmitted to recording/reproducing unit 140 has a logical data structure shown in Figure 5A. (EX1006, 4:46-48.)

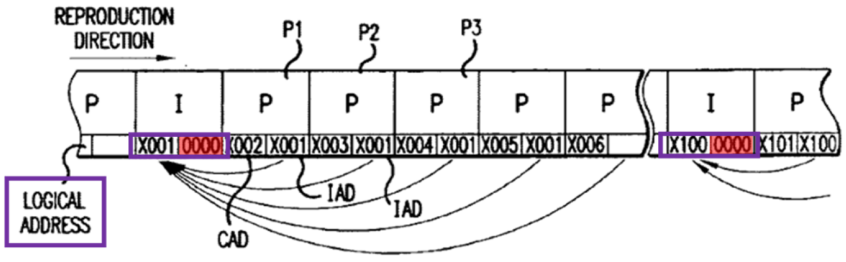


FIG.5A

(EX1006, Fig.5A.) And Kim refers to decoding image data as "reproducing." (*Id.*, 4:46-48; EX1003, ¶¶153, 158.) When Kim's decoder "reproduces" the bitstream, it resets the IAD component of each logical address for an I-picture as shown to preserve the logical structure produced during encoding. Were it otherwise, the decoder could not successfully reproduce the original bitstream. (EX1003, ¶¶157-58.)

3. Claim 10

Claim 10 is substantively the same as claim 8, except it recites *a video decoder comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the video decoder to decode a compressed video sequence*.

a. 10[pre]

To the extent limiting, Kim teaches or suggests 10[pre]. As discussed in §VII.B.2.a for 8[pre], Kim teaches *decoding a compressed video sequence*. Kim also teaches the rest of 10[pre].

Kim discloses a Time Lapse recording/reproduction device that includes "a decoder 111 for performing a reverse-quantization and reverse-transformation of…demultiplexed image data"; "a reproducing-DSP 121 for generating original image frames based on the P-picture data output from the decoder 111 and moving vector output from the de-MUX 131"; and "a controller 150 for controlling the above-described components based on the commands from the user." (EX1006, 3:18-27, Fig. 2.) A POSITA would have understood that the controller and corresponding decoder and reproducing DSP comprise *at least one processor and at least one memory including computer program code configured to, with the at least one processor, cause the video decoders to decode a compressed video sequence*. (EX1003, ¶¶161-62.) For example, a DSP is a specialized type of computer processor that executes a stored program of computer instructions for performing signal processing operations. (EX1003, ¶162.) Thus, Kim's disclosure of the DSP by itself is sufficient.

Furthermore, Kim discloses encoding image frames in accordance with MPEG (EX1006, 4:15-19) and subsequently decoding those image frames (*id.*, 5:8-

68). As noted in §VII.A.3.a, it would have been obvious to a POSITA to implement an MPEG-1 decoder as *a video decoder comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the video decoder to decode a compressed video sequence*. Kim discloses decoding video that conforms to the MPEG standard. Accordingly, for the reasons discussed in §VII.A.3.a, it would have been obvious to a POSITA to implement Kim's decoder as *a video decoder comprising at least one processor and at least one memory including computer program code, the at least one memory and the computer program code configured to, with the at least one processor, cause the video decoder to decode a compressed video sequence*.

b. 10[a]

This limitation is substantively identical to 8[a], and Kim teaches it for the same reasons. §VII.B.2.b.

c. 10[b]

This limitation is substantively identical to 8[b], and Kim teaches it for the same reasons. §VII.B.2.c.

d. 10[c]

This limitation is substantively identical to 8[c], and Kim teaches it for the same reasons. §VII.B.2.d.

e. 10[d]

This limitation is substantively identical to 8[d], and Kim teaches it for the same reasons. *See* §VII.B.2.e.

Claim 10 is thus unpatentable as obvious over Kim.

4. Claim 11

Claim 11 is substantively the same as claim 8, except it recites *a computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence*.

a. 11[pre]

To the extent limiting, Kim teaches or suggests 11[pre]. (EX1003, ¶¶164-69.) As discussed in §VII.B.2.a for 8[pre], Kim teaches *decoding a compressed video sequence*. Kim teaches the additional elements of 11[pre].

Kim's Time Lapse recording/reproduction device includes "a decoder 111 for performing a reverse-quantization and reverse-transformation of…demultiplexed image data"; "a reproducing-DSP 121 for generating original image frames based on the P-picture data output from the decoder 111 and moving vector output from the de-MUX 131"; and "a controller 150 for controlling the above-described components based on the commands from the user." (EX1006, 3:18-27.)

A POSITA would have understood these components to be *data processing devices* given Kim's disclosure that they all process data in the form of image data or commands. (EX1003, ¶164.)

And as Dr. Saber explains, a POSITA would have found it obvious to implement Kim's decoder 111, reproducing DSP 121, and controller 150 as *a computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence*. (EX1003, ¶¶165-69.) For example, because a DSP is a specialized type of processor that executes a stored program of computer instructions for performing signal processing operations, a POSITA would have understood that Kim's reproducing DSP 121 necessarily performs its portion of the decoding process by executing a *computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence*. (EX1003, ¶169.) Likewise, a POSITA would have found it obvious to implement the remaining video decoding functions performed by Kim's decoder 111 and controller 150 as a computer program running on a general-purpose processor that was available at the time. (EX1003, ¶¶167-68.) Finally, a POSITA would have understood that these computer program products would have been *stored on a non-transitory computer readable medium* so that they

Nokia Exhibit 2012, p.74
ASUS v. Nokia
IPR2025-01279

could be stored and then executed by the one or more processors to decode a video sequence based on the commands from the user. (EX1003, ¶168.)

Furthermore, Kim discloses encoding image frames in accordance with MPEG (EX1006, 4:15-19) and subsequently decoding those image frames (EX1006, 5:8-68). As noted above in §VII.A.4.a, it would have been obvious to a POSITA to implement an MPEG-1 decoder as *a computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence*. Thus, for the reasons discussed in §VII.A.4.a, it would have been obvious to a POSITA to implement Kim's decoder as *a computer program product, stored on a non-transitory computer readable medium and executable in a data processing device, for decoding a compressed video sequence*. (EX1003, ¶169.)

b. 11[a]

This limitation is substantively identical to 8[a], except it recites *a computer program code for* decoding an indication rather than the act of decoding an indication. As explained in §VII.B.2.b, Kim teaches *decoding from the video sequence an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence, wherein all motion-compensated temporal prediction references of the independent sequence refer only to image frames within said independent sequence*. And as explained in §VII.B.4.a, a

Nokia Exhibit 2012, p.75
ASUS v. Nokia
IPR2025-01279

POSITA would have recognized that the decoding would be carried out by *a computer program product*, which would include *computer program code*. Kim, therefore, teaches or suggests 11[a].

c. 11[b]

This limitation is substantively identical to 8[b], except it recites *a computer program code for* starting the decoding rather than the act of starting the decoding. As explained in §VII.B.2.c, Kim teaches *starting the decoding of the video sequence from said first image frame of the independent sequence, whereby the video sequence is decoded without prediction from any image frame decoded prior to said first image frame*. And as explained in §VII.B.4.b, a POSITA would recognize that the decoding process would be carried out by *a computer program product*, which would include *computer program code*. Kim, therefore, teaches or suggests 11[b].

d. 11[c]

This limitation is substantively identical to 8[c], except it recites *a computer code for* decoding identifier values rather than the act of decoding identifier values. As explained in §VII.B.2.d, Kim teaches *decoding identifier values for image frames according to a numbering scheme*. And as explained in §VII.B.4.a, a POSITA would recognize that the decoding process would be carried out by *a computer program product*, which would include *computer program code*. Kim, therefore, teaches or suggests 11[c].

e. 11[d]

This limitation is substantively identical to 8[d], except it recites *a computer program code for* resetting the identifier value rather than the act of resetting the identifier value. As explained in §VII.B.2.e, Kim teaches *resetting the identifier value for the indicated first image frame of the independent sequence*. And as explained in §VII.B.4.a, a POSITA would recognize that the decoding process would be carried out by *a computer program product*, which would include *computer program code*. Kim, therefore, teaches or suggests 11[d].

For the reasons discussed above, claim 11 would have been obvious over Kim.

C. Ground 3: MPEG-1 and Yagasaki Render Obvious Claim 9

1. Yagasaki

Yagasaki discloses methods and apparatuses for encoding and decoding an image signal. (EX1015, Title.) Yagasaki describes MPEG as a "highly efficient encoding system[] which rel[ies] upon the correlation of video signals." (EX1015, 1:36-39.) A POSITA would have understood that Yagasaki refers to the same "MPEG" described by MPEG-1. (EX1003, ¶¶170-71.)

Further, Yagasaki describes a video signal encoded by an MPEG system as having a "video sequence…divided into a frame group (GOP)," where each of the frames is classified as an I-picture, P-picture, or B-picture. (EX1015, 1:58-67.) Consistent with MPEG-1, Yagasaki describes an I-picture as an "Intra-coded frame

and "the only type of frame that can be directly accessed and decoded" without reference to other frames. (EX1015, 1:63, 2:10-17.)

According to Yagasaki, hardware limitations make it difficult to "select a frame freely and reproduce it at a high speed." (EX1015, 2:14-15, 23-24.) This is because P- and B-pictures "cannot be decoded in the absence of decoded images of past and/or future frames." (EX1015, 2:12-13.) Even I-pictures "can be reproduced at high speed…only with rough motion." (EX1015, 2:19-20.)

Yagasaki overcomes these problems by providing a "decoding apparatus capable of reproducing a moving image smoothly at a high speed." (EX1015, 2:29-32.) It achieves this advantage by providing at least some predictive-coded pictures (P- or B-pictures) with designated slices that "consist entirely of intra-coded macroblocks." (EX1015, 2:41-44.) "An indication of the slices designated is provided in the coded picture signals." (EX1015, 2:44-45.)

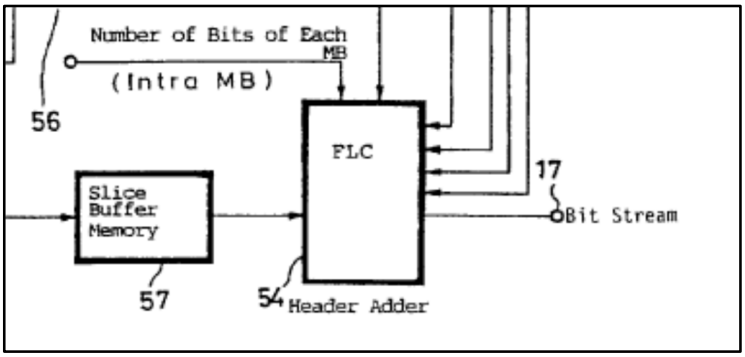
2. Claim 9

MPEG-1 discloses *a method according to claim 8*, which in turn recites *a method for decoding a compressed video sequence*, for the reasons described above in §VII.A.2.a. Yagasaki discloses the rest of claim 9. (EX1003, ¶¶174-80.)

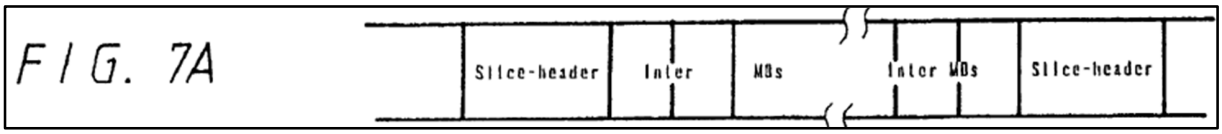
First, Yagasaki discloses that an image may be formed of multiple slices composed of "macroblocks," each of which is *a unit range to be encoded*. (EX1015, 9:27-33.) A POSITA would have recognized that the slices described by Yagasaki

are also described in MPEG-1. (*See, e.g.*, EX1005, 26 ("slice [video]: A series of macroblocks. It is one of the layers of the coding syntax defined in [MPEG-1]."); EX1003, ¶175.)

Yagasaki also discloses slice headers. It describes a "hybrid encoder 102" that includes a "fixed length coder" (FLC) 54 that Yagasaki refers to as a "header adder" for adding a header to image data stored in slice buffer memory 57.

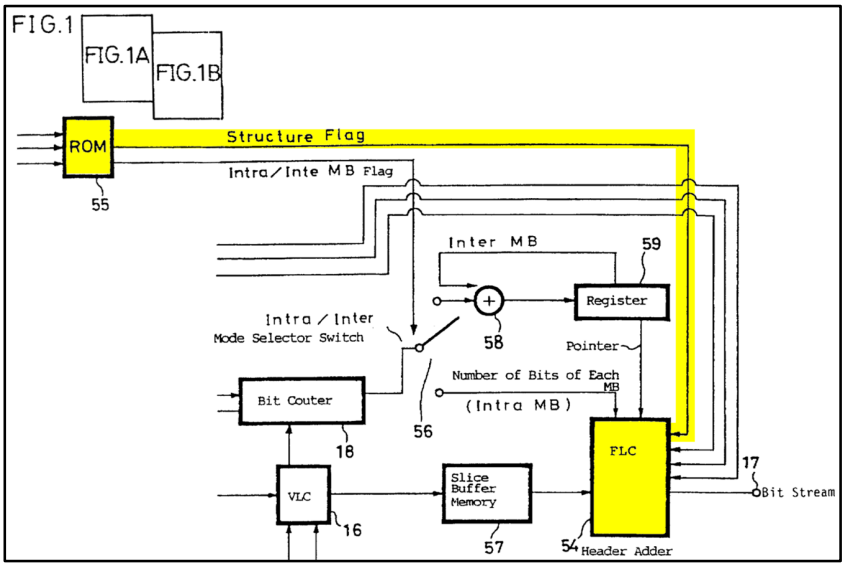


(EX1015, 4:56, 6:15-24, 6:39-41, 7:3-4, Fig. 1B (excerpt).) As Yagasaki claim 38 recites, the "header adder" is "for adding slice headers." (EX1015, 23:4-5.) Yagasaki further shows slices containing these slice headers:



(EX1015, Fig.7A.) A POSITA would have recognized that the slice header described by Yagasaki is also described in MPEG-1. (EX1005, 91-92 (describing slice headers); EX1003, ¶176.)

The information added to the slice header includes a separate flag. A "ROM 55 holds a distribution pattern of intraframe coding macroblocks for high-speed reproduction which are assigned to each of the frames of the GOP." (EX1015, 6:42-44.) "[T]he ROM 55 outputs, to a macroblock mode selector switch 56, a flag which indicates whether a structure flag of the slice and the macroblock are an intra-frame coding macroblock for high-speed reproduction." (EX1015, 6:44-50.) "[T]he intra-frame coded macroblocks can be extracted using only the flag information in the slice header, without decoding the entire bit stream." (EX1015, 10:67-11:3.) As shown below, ROM 55 passes the structure flag to FLC 54, which in response "adds information to a slice header." (EX1015, 7:1-4.)



(EX1015, Figs. 1A, 1B (excerpt).)

Information added to the slice header includes "a flag representing a structure of a slice." (EX1015, 9:55-61; EX1003, ¶177.) This separate flag renders obvious

an indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence. Specifically, the flag is "indicative of whether there is an intra-frame coded macroblock to be decoded within any particular slice upon reproduction at a high speed in both forward and reverse directions, and is also indicative of the structure of such a macroblock if it exists." (EX1015, 10:1-5.) The slice may be "entirely composed of intra-frame coded macroblocks." (EX1015, 10:14-15.) A POSITA would have recognized that a frame composed of such slices functions as the beginning of *an independent sequence*. (EX1015, 2:7-9 (I-picture is intra-coded in its entirety); EX1003, ¶¶178-80.) Accordingly, Yagasaki teaches a method of decoding a compressed video sequence that includes *separate flags included in the header of a slice*, which is part of a frame. These flags identify when a slice is composed entirely of intra-frame coded macroblocks. (EX1015, 10:26.)

Reading Yagasaki, a POSITA would have found it obvious to place flags marking intra-coding status in slice headers. Because a fully intra-coded frame is the beginning of an independent sequence, it would also have been obvious to a POSITA that flags indicating this same characteristic from the MPEG-1 ground (closed_GOP or Temporal Reference) would similarly be included in a slice header. (EX1003, ¶180.)

For these reasons, claim 9 would have been obvious over the combination of MPEG-1 and Yagasaki.

3. Motivation to Combine MPEG-1 and Yagasaki

A POSITA would have been motivated to implement the teachings of Yagasaki within the encoding and decoding methods disclosed by MPEG-1. (EX1003, ¶¶181-86.) Yagasaki provides video coding methods that may be performed in an MPEG-1 system, with predictable results. Indeed, Yagasaki expressly discloses that the "basic coding process in the present invention is the same as the encoding process in MPEG." (EX1015, 7:5-6.) Additionally, Yagasaki describes the same GOPs and I-pictures taught in MPEG-1. (EX1015, 1:58-2:9; EX1005, 71.) I-pictures are intra-coded in their entirety and thus contain slices with all intra-coded macroblocks; Yagasaki and MPEG-1 both confirm these are the only kind of frames that may be directly accessed and decoded independently. (EX1015, 2:7-18; EX1005, 23, 71.) Yagasaki's decoding process for full intra-coded slices may also be performed in an MPEG-1 system—Yagasaki explains that slices "with all intra macroblocks…can be decoded by the normal MPEG decoding process." (EX1015, 17:32-36.)

Yagasaki further recognizes the desirability of efficient video coding, which reduces recording size. (EX1015, 1:31-36.) Yagasaki notes that frames cannot be entirely intra-coded I-pictures; they require too much time to read and process. (EX1015, 2:16-28.) But Yagasaki teaches that intra-frame coded regions are critical for decoding, must "cover at least the entire image area" in one GOP, and should not

"leave gaps on the image area." (EX1015, 8:22-28.) A POSITA would have understood this is because non-intra-frame coded regions do not contain sufficient data to decode a full image. (EX1003, ¶182.)

For "high-speed reproduction" during decoding, Yagasaki describes outputting only slices containing intra-frame coded macroblocks to a buffer while discarding others. (EX1015, 14:30-40.) Thus, a POSITA would have recognized that an indication of the intra-coding status for slices, as taught by Yagasaki, allows for efficient decoding. (EX1003, ¶183.) Yagasaki teaches doing so using separate flags in the slice headers, as discussed in §VII.C.2.

A POSITA would have further been motivated to use the indication method taught by Yagasaki to provide other indications of intra-coding status disclosed by MPEG-1 for other image regions, including closed_GOP or Temporal Reference. (EX1003, ¶184.) The modification would have improved the efficiency of decoding by identifying for the decoder the intra-coded image regions that provide the underlying image data from which the decoder may derive the remaining images of a GOP. (EX1003, ¶184.)

Moreover, a POSITA would have found it obvious to try to include the indications of intra-coding status disclosed by MPEG-1 in the slice header. There are only a finite number of header locations in which such indications can be placed, and a POSITA would have recognized that the slice header is one of those locations.

Further, a POSITA would have had a reasonable expectation of success that placing indications of intra-coding status in the slice headers would be successful. Yagasaki teaches that such notifications can be placed in slice headers and successfully decoded using the normal MPEG-1 decoding process. (EX1003, ¶¶185-86.)

A POSITA would thus have been motivated to modify the methods disclosed by MPEG-1 using these teachings of Yagasaki.

VIII. CONCLUSION

Inter Partes Review of the Challenged Claims is respectfully requested.[13]

IX. MANDATORY NOTICES

A. Real Party in Interest

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

---

[13] Currently, the memorandum on interim processes for PTAB workload management applies, and the Office has stated the petition should not address discretionary issues; instead, those issues should be raised in opposition to PO's discretionary denial brief (if filed). *See Interim Processes for PTAB* Coke Stewart, *Workload Mgmnt.*, (Mar. 26, 2025), https://www.uspto.gov/patents/ptab/faqs/interim-processes-workload-management.

Nokia Exhibit 2012, p.84
ASUS v. Nokia
IPR2025-01279

B. Related Matters

To the best of Petitioners' knowledge, the '321 patent has been involved in the following related matters:

- *Nokia Technologies Oy v. Amazon.com, Inc.*, 1:23-cv-01232 (D. Del.)
- *Nokia Corporation v. Amazon.com, Inc.*, 1:23-cv-01236 (D. Del.)
- *Nokia Technologies Oy v. HP Inc.*, 1:23-cv-01237 (D. Del.)
- *Certain Video Capable Electronic Devices, Including Computers, Streaming Devices, Televisions, and Components and Modules Thereof*; Inv. No. 337-TA-1380 (ITC)
- *Amazon.com, Inc., Amazon.Com Services, LLC v. Nokia Technologies OY*, IPR2024-00691

C. Notice of Counsel and Service Information

LEAD COUNSEL
Jessica Kaiser (Reg. No. 58,937) kaiser-ptab@perkinscoie.com Perkins Coie LLP 1900 Sixteenth Street, Suite 1400 Denver, CO 80202-5255 Telephone: (303) 291-2300
BACK-UP COUNSEL
Jon R. Carter, Reg. No. 75,145 PERKINS COIE LLP 1155 Avenue of the Americas 22nd Floor New York, NY 10036

Tel: (212) 262-6900
Carter-ptab@perkinscoie.com

Respectfully submitted,

/ *Jessica Kaiser* /

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

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

Date: June 4, 2025

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

Pursuant to 37 C.F.R. §42.24(a), Petitioners hereby certify that portions of the above-captioned Petition for *Inter Partes* Review of U.S. Patent 8,050,321, in accordance with and reliance on the word count provided by the word-processing system used to prepare this Petition, that the number of words in this paper is 13,987. Pursuant to 37 C.F.R. §42.24(a), this word count is in compliance and excludes the table of contents, table of authorities, mandatory notices under §42.8, certificate of service, certificate of word count, appendix of exhibits, and any claim listing. This word count was prepared using Microsoft Word.

Respectfully submitted,

/ *Jessica Kaiser* /

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

Date: June 4, 2025
PERKINS COIE LLP
1900 Sixteenth Street, Suite 1400
Denver, CO 80202-5255

CERTIFICATE OF SERVICE

The undersigned hereby certifies that true copies of the Petition for *Inter Partes* Review of U.S. Patent No. 8,050,321 and supporting materials (Exhibits and Power of Attorney) were served via overnight delivery on the Patent Owner at the correspondence address of record as listed on PAIR:

Nokia
c/o Mueting Raasch Group
111 WASHINGTON AVE. S., SUITE 700
Minneapolis, MN 55401

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

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

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

Respectfully submitted,

/ *Jessica Kaiser* /

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

Date: June 4, 2025