UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ASUSTEK COMPUTER INC.,

Petitioner

v.

NOKIA CORP. AND NOKIA TECHNOLOGIES OY,

Patent Owner

Case: IPR2025-01279

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

PATENT OWNER'S PRELIMINARY RESPONSE

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

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Page(s)

CASES

STATUTES

TABLE OF EXHIBITS

Ex. No.	Title	Previously Submitted
2001	Order No. 10-First Amended Adopted Procedural Schedule in Investigation No. 337-TA-1448, dated June 25, 2025	**X**
2002	Respondents' First Supplemental Invalidity Contentions in Investigation No. 337-TA-1448, dated August 8, 2025	**X**
2003	Respondents' Exhibit B-5 Claim Chart (U. S. Patent No. 6,912,351 ("Kim")) in Investigation No. 337-TA-1448	**X**
2004	Respondents' Exhibit B-4 Claim Chart (ISO/IEC 11172 ("MPEG-1")) in Investigation No. 337-TA-1448	**X**
2005	Respondents Joint Disclosure of Initial Invalidity Contentions in Response to Individual Interrogatory Nos. 10-12 in Investigation No. 337-TA-1380, dated February 15, 2024	**X**
2006	Respondents' Exhibit B-5 Claim Chart (U. S. Patent No. 6,912,351 ("Kim") in Investigation No. 337-TA-1380	**X**
2007	Respondents' Exhibit B-8 Claim Chart (ISO/IEC 11172 ("MPEG-1")) in Investigation No. 337-TA-1380	**X**
2008	Public version of Initial Determination on Violation of Section 337 and Recommended Determination on Remedy and Bond in Investigation No. 337-TA-1380, dated December 20, 2024	**X**
2009	Complaint Under Section 337 of the Tariff Act of 1930, As Amended, in Investigation No. 337-TA-1448, dated June 25, 2025	**X**
2010	Declaration of Dr. Eli Saber submitted in IPR2025-01068	**X**
2011	Original Complaint in *Nokia Technologies Oy v. Asustek Computer Inc. et al.*, Case No. 2:25-cv-03053 (C.D. Cal.), April 7, 2025	**X**
2012	*Element TV Company, LP, and Element Television Company, LLC v. Nokia Technologies Oy*, IPR2025-01068, Paper 1 (PTAB July 16, 2025)	**X**
2013	*Element TV Company, LP, and Element Television Company, LLC v. Nokia Technologies Oy*, IPR2025-01068, Paper 7 (PTAB Aug. 25, 2025)	**X**

2014	*Element TV Company, LP, and Element Television Company, LLC v. Nokia Technologies Oy*, IPR2025-01068, Paper 9 (PTAB Aug. 27, 2025)	**X**
2015	Declaration of Dr. Iain Richardson	
2016	CV of Dr. Iain Richardson	
2017	ITU-T Rec. H.264 (03/2005) ("H.264")	
2018	Public version of Initial Determination on Violation of Section 337 and Recommended Determination on Remedy and Bond in Investigation No. 337-TA-1380, dated December 20, 2024 ("1380 Public ID")	
2019	*Certain Video Capable Electronic Devices, Including Computers, Streaming Devices, Televisions, and Components and Modules Thereof*, Investigation No. 337 TA-1380 (U.S.I.T.C.), Excerpts of Evidentiary Hearing Transcript (Public), dated Sept. 9-13, 2024 ("1380 Public Tr.")	
2020	Iain E. Richardson, *The H.264 Advanced Video Compression Standard* (2d ed. 2010) ("Richardson, H.264")	
2021	Hannuksela, "Simple Definition of GOP for Random Access," Joint Collaborative Team (JVT) of ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, 2nd Meeting, Geneva, CH, Jan. 29 - Feb. 1, 2022, Document JVT-B041 ("JVT-B041")	

I. INTRODUCTION

The Board should deny institution because Petitioner fails to show a reasonable likelihood that claims 8-11 (the "Challenged Claims") of U.S. Patent No. 8,050,321 (the "'321 Patent") are obvious.

The '321 Patent is directed to novel data structures and techniques that enable a user to watch a streaming video from a random user-selected point in the stream, which is sometimes called "random access." The '321 Patent provides random access of streaming video with more efficiency and fewer errors than conventional technology at the time.

MPEG-1 does not invalidate Claims 8, 10, or 11 of the '321 Patent. Petitioner relies on the MPEG-1 video coding standard to attempt to show that the inventions of the '321 Patent are not novel, and, despite numerous evolutions of video coding with working groups of experts, the claims of the '321 Patent are nothing more than what was done in MPEG-1. This is wrong, for numerous reasons.

Most devastating to Petitioner's argument is that MPEG-1 does not disclose the [d] limitations of the Challenged Claims—the decoder "resetting the identifier value" limitation. The reason is simple. Petitioner has failed to identify ***any*** disclosure in MPEG-1 that shows the ***decoder*** resetting the identifier value. Instead, Petitioner attempts to point to an identifier value of the *next* Group of Picture ("GOP")—one which the ***encoder*** sets to zero for each new GOP—to argue that the identifier value of the current GOP is ***reset*** to zero by a ***decoder***. Petitioner's argument points to the wrong process of video coding (the encoder instead of the decoder) and to the wrong parameter.

Petitioner fully understands this flaw. To save the reference, Petitioner argues that when the decoder merely reads off the values already set by the encoder, the decoder itself does the resetting of the value. Petitioner does not disagree that when a new GOP is reached in display order, the *encoder* sets the Temporal Reference for the first frame of the GOP in display order. What Petitioner erroneously argues is that, when that same frame is reached in the decoding process, the decoder merely reproducing the flag set by the encoder constitutes "resetting" the flag.

But this flaw is not the only one. Petitioner also argues that the [a] limitations (which require an indication of a first image frame in decoding order) is met by the closed_GOP flag that is a syntax element within the Group of Pictures header. This header precedes any image frame in the bitstream. And closed_gop flag does not confirm the presence of any image frame to a decoder. Rather, closed_gop flag provides information only about the group of pictures layer, not any information about any picture. To circumvent this fact, Petitioner and its expert craft a modified MPEG-1 decoder that excludes any possibility that the decoder would ever successfully decode bitstreams other than ones with an I-frame after every GOP header. As a threshold matter, this argument confuses what a POSITA would understand an "indication" to be. In video coding, an indication confirms something to a decoder with certainty. In Petitioner's modified decoder that it posits would encounter only a controlled universe of bitstreams, the fact that an I-frame follows a GOP header does not mean the GOP header indicated the I-frame. Rather, even in Petitioner's contrived scenario, the modified decoder would only confirm the presence of an I-frame in the bitstream through picture layer syntax (not the closed_gop flag). Furthermore, Petitioner's closed_gop flag can only signal the possibility of and cannot confirm (i.e., indicate) an I-frame.

Petitioner's next argument is that the [a] limitations are met by the Temporal Reference flag, a flag that tracks the display order of frames, not the decoding order of frames. Petitioner cherry picks an example where the first frame of the display order happens to be the first frame of the decoding order, arguing that this coincidence in indexing satisfies the limitation requiring that the first image in decoding order be indicated. In doing so, Petitioner ignores all instances in which the first image in display order is not the first image in decoding order. In all of these instances, Temporal Reference does *not* indicate "the first image frame, in decoding order" that the claims specify. This is important, because these examples make clear that a Temporal Reference of zero (as the name would suggest) indicates the first image in display order, not decoding order.

It is unsurprising that Petitioner's arguments on MPEG-1 are flawed. The '321 Patent sought to improve prior art random access, including the limited random access provided by MPEG-1's Group of Pictures. What the '321 Patent inventor sought to improve was a numbering scheme to facilitate random access within any independent sequence and not to confuse the decoder with numbering schemes that would make it appear as some frames were lost (e.g. starting at frame 15 in display order or decoding order, where MPEG-1 would attempt to reconstruct frames 1-14). In resetting the identifier value to zero at the decoder, the decoder can then process the independent sequence without error. Put simply, the '321 Patent improved the very random access techniques used in MPEG-1 that Petitioner incorrectly alleges disclose the claims of the '321 Patent.

Kim does not invalidate Claims 8, 10, or 11 of the '321 Patent. Kim—which is not about random access or streaming video, but is instead about storing time lapse intermittent images on an optical disk to improve durability over traditional magnetic storage—fails to disclose or render obvious numerous elements of the Challenged Claims. Most significantly, Kim does not disclose or render obvious decoding the required "***indication*** of at least one image frame, which is the ***first image frame, in decoding order***, of the independent sequence." ASUS-1001, 8-11 (decoding). This is not surprising given that Kim does not discuss independently decodable video sequences anywhere in its disclosure but instead emphasizes the problem of decoding *individual image frames* that are intermittently stored in the bitstream. The Petition relies on an optional value that Kim describes may be "zero" or entirely "absent" from the bitstream because it serves no purpose at all. This optional value is never ascribed any meaning and is never used by either the encoder or decoder as an indication of anything, which is why it is allowed to be entirely absent from the bitstream. The mere fact that a zero value may sometimes be present in an image that may sometimes be the first image frame in decoding order does not mean that the zero value *indicates* the first image frame in decoding order, as the claims require. Indeed, the fact that a zero value may also be present in an image that is *not* the first image frame in decoding order proves that the zero value cannot be the claimed "***indication*** of at least one image frame, which is the ***first image frame, in decoding order***, of the independent sequence." The Petition does not present any theories on how or why a person of ordinary skill would have modified Kim to overcome this deficiency or render obvious this limitation.

Additionally, Kim does not disclose or render obvious the *decoder* resetting the identifier value for the indicated first image frame of the independent sequence (claims 8-11). This is because Kim at best only describes resetting values at the *encoder*. Just as in MPEG-1, the decoder takes no role in resetting any identifiers—as Petitioner recognizes, the decoder's sole job here is to reproduce the bitstream faithfully. For this additional reason, Kim fails to disclose or render obvious the Challenged Claims.

Notably, the patentability of the ’321 Patent in view of Kim, and, separately, in view of MPEG-1, has already been adjudicated in an ITC proceeding in 2024. There, the ALJ determined that the same Kim and MPEG-1 references did not anticipate or render obvious any of the asserted decoding claims of the ’321 Patent. EX-2018, 49-56, 56-57.

Although the Board has instituted an IPR on the ’321 Patent in the past (IPR2024-00691), that petition differed materially from the Petition at issue here, and the Board did not at that time have the benefit of the evidence, testimony, and findings in the related ITC Investigation that showed the Challenged Claims are valid over Kim and MPEG-1. This response also focuses on specific admissions by Dr. Saber that were not focused on in the prior IPR.

Therefore, for the reasons explained herein, Nokia requests the Board find all the Challenged Claims valid, just as the ALJ did for the decoding claims challenged here.

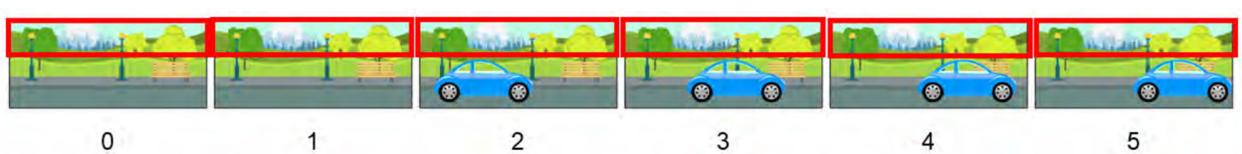
II. BACKGROUND

A. Technical Background

The ’321 Patent relates to improvements to video encoders and decoders, which enables a decoder to identify where to start a decoding process, while minimizing errors and optimizing efficiency. This is particularly important for situations when a user wishes to select a particular point (other than the beginning) to start watching a video, for example when joining a stream part-way through or when skipping ahead or returning to a past point in the video. In video coding, this is known as random access, and the '321 Patent is directed to improving random access. ASUS-1001, 3:52-56, 3:62-4:4, 4:48-60.

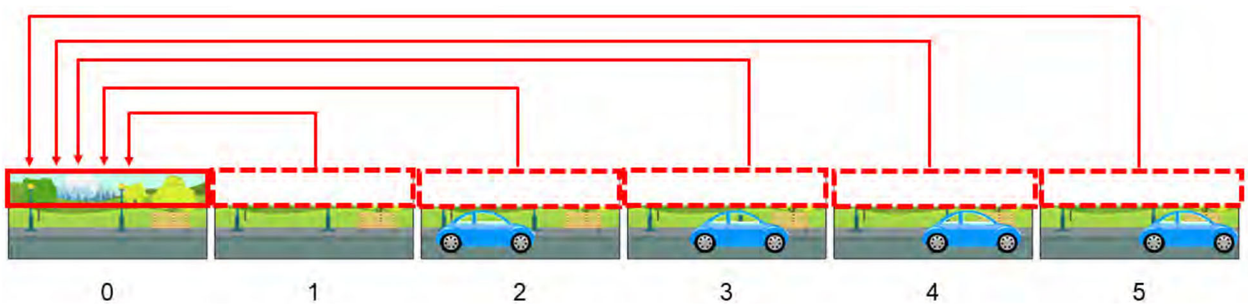
Minimizing bitrate is extremely important for applications such as video streaming, and various techniques have been developed to improve video compression (video encoding), such as techniques that take advantage of spatial redundancy (similarities between areas within a video frame known as intra-frame techniques) and temporal redundancy (similarities between successive video frames known as inter-frame techniques). As the '321 Patent notes, "[t]o reduce the amount of data in video files, the image data can be compressed into a smaller form by reducing the amount of redundant information in the image frames." ASUS-1001, 2:5-7.

One such inter-frame technique, known as motion-compensated temporal prediction, can be used to take advantage of the redundancy of successive image frames. ASUS-1001, 2:15-17.



EX-2015, ¶55 (annotating in red the redundancy in successive image frames).

In this method, a portion of a current frame to be encoded can be predicted from a portion in another, previously-encoded frame such as a past or future frame in display order. Motion-compensated temporal prediction attempts to identify a matching portion in a previously coded frame, which may be in a different position due to movement or motion of objects or areas, and to use this to predict a portion of a current frame. ASUS-1001, 2:16-20.



EX-2015, ¶57 (annotating in red how successive image frames can reuse the redundant portion of the previous frame rather than re-encode that portion of the image).

I-frames (or INTRA-frames) are encoded frames whose encoded image information is not determined using this motion-compensated temporal prediction process, while INTER-frames (also known as P-frames or B-frames) are encoded frames whose image information is, at least in part, determined based on prediction from one or more previously coded frames in the sequence. In the example shown above, the 0 frame is an I-frame, and the other frames 1-5 are P-frames. Such previously coded frames may be preceding frames in display order and/or subsequent frames in display order. A P-frame is encoded using only one previously encoded frame and a B-frame may be encoded using two previously-encoded frames, for example one preceding frame and one subsequent frame in display order. ASUS-1001, 2:20-32, 2:42-47.

While the use of INTER-frames, such as P-frames, improves coding efficiency, one problem that the '321 Patent describes is that "[i]f a frame is lost, frames dependent on it can no longer be correctly decoded." ASUS-1001, 2:31-33. The '321 Patent describes that, in conventional systems, image "frames are typically numbered according to an arithmetical series," so that the decoder can know if a frame is missing based on a discontinuity in the frame numbering. ASUS-1001, 3:19-20, 11:20-25. However, in conventional systems, the decoder has no way of knowing whether lost or missing frames were needed for further decoding or whether they could be ignored. ASUS-1001, 11:22-25, 3:3-20.

One situation where this was a problem is "when the user wishes to start browsing a video file in the middle of a video sequence. Such situations occur for example when the user wishes to browse a locally stored video file backward or forward or to browse a streaming file at a particular point; when the user initiates the playback of a streaming file from a random point; or when a video file that is to be reproduced is detected to contain an error that interrupts the playback or requires the playback to be resumed from a point following the error. When the browsing of a video file is resumed from a random point after previous browsing, discontinuity typically occurs in the image numbering. The decoder typically interprets this as unintentional loss of image frames and unnecessarily tries to reconstruct the image frames suspected as lost." ASUS-1001, 11:11-25.

"Another example relates to starting the reception of a broadcast or multicast video transmission from the middle of the video transmission. A third example relates to on-demand streaming from a server and occurs when an end-user wishes to start playback from a certain position of a stream." ASUS-1001, 3:66-4:5.

Thus, the inventor the '321 Patent recognized that there was a need for an "improved method and equipment implementing the method, that enable detection of image frames from which a decoder can start the decoding process" without having to worry about whether the missing frames are needed for decoding later frames. ASUS-1001, 4:8-30; EX-2015, ¶65. That is, it is important to be capable of identifying a coded video frame (picture) from which it is guaranteed that a decoder can begin decoding independently, for example to enable playback from or seeking to start at a particular position in a coded video stream (random access). EX-2015, ¶65.

B. The ’321 Patent

The ’321 Patent claims an improved method “that enable[s] detection of image frames from which a decoder can start the decoding process.” ASUS-1001, 4:8-10. The inventor of the ’321 Patent recognized that the problems with conventional systems “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.” ASUS-1001, 11:26-30.

The idea behind the ’321 Patent’s invention was to be able to identify during later *decoding* of a bitstream an initiation picture among a series of pictures in an independently decodable sequence to begin decoding the bitstream from that picture; and thus, during the decoding process, any picture before the initiation picture is discarded, by being defined as unusable as a reference picture. *See* ASUS-1001, 4:31-35. “Consequently, when activated at a random point, the decoder preferably sets the identifier of the first image frame, preferably an I-frame, of the independently decodable sub-sequence at zero.” ASUS-1001, 11:26-40. To that end, the ’321 Patent claims a method of decoding comprising “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; ***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; decoding identifier values for image frames according to a numbering scheme; ***and resetting the identifier value for the indicated first image frame*** of the independent sequence." ASUS-1001, claim 8 (emphasis added).

C. The Prior Art

1. MPEG-1 (ISO/IEC 11172)

MPEG-1 Video (ISO/IEC 11172-2, "MPEG-1") is a standard developed by the Moving Pictures Expert Group that was approved in 1993. MPEG-1 does not enable random access at a user-specified playback point and thus did not solve the prior-art problem identified and claimed by the '321 Patent. Instead, MPEG-1 describes that after a user specifies a random access point, the decoder looks for the next sequence start code, group of pictures (GOP) start code, or picture start code that introduces an I-picture (collectively, "start codes") to start decoding. EX-1005, 43, 87, 120; EX-2015, ¶¶85-90. Thus, in MPEG-1, decoding and playback starts from the next start code rather than from a specific user-identified frame (i.e., the random access point). *Id.*

Several properties of groups of pictures in MPEG-1 are important to keep in mind. MPEG-1 defines a Group of Pictures or GOP below:

D.5.2 Group of pictures

Two distinct picture orderings exist, the display order and the bitstream order (as they appear in the video bitstream). A group of pictures (gop) is a set of pictures which are contiguous in display order. A group of pictures must contain at least one I-picture. This required picture may be followed by any number of I and P-pictures. Any number of B-pictures may be interspersed between each pair of I or P-pictures, and may also precede the first I-picture.

Property 1. A group of pictures, 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.

Property 2. Another property of a group of pictures is that it must begin, in display order, with an I or a B-picture, and must end with an I or a P-picture. The smallest group of pictures consists of a single I-picture, whereas the largest size is unlimited.

The original concept of a group of pictures was a set of pictures that could be coded and displayed independently of any other group. In the final version of this part of ISO/IEC 11172 this is not always true, and any B-pictures preceding (in display order) the first I-picture in a group may require the last picture in the previous group in order to be decoded. Nevertheless encoders can still construct groups of pictures which are independent of one another. One way to do this is to omit any B-pictures preceding the first I-picture. Another way is to allow such B-pictures, but to code them using only backward motion compensation.

Property 3. From a coding point of view, a concisely stated property is that a group of pictures begins with a group of pictures header, and either ends at the next group of pictures header or at the next sequence header or at the end of sequence, whichever comes first.

Some examples of groups of pictures are given below:

I

I P P

I B P B P

B B I B P B P

B B I B B P B B P B B P

B I B B B B P B I B B I I

Figure D.17 -- Examples of groups of pictures in display order

These examples illustrate what is possible, and do not constitute a suggestion for structures of groups of pictures.

EX-1005, 84-85 (explaining and providing examples of Group of Pictures as defined in MPEG-1); EX-2015, ¶¶71-72.

Petitioner's arguments refer to MPEG-1's "closed_gop flag." MPEG-1's closed_gop flag is part of the "Group of pictures layer." That is, closed_gop flag is part of the syntax layer that applies to an entire GOP. EX-1005, at 43.

closed_gop -- This is a one-bit flag which may be set to "1" if the group of pictures has been encoded without motion vectors pointing to the previous group of pictures.

See EX-2015, ¶75.

As specified in MPEG-1, the closed_gop flag is received in the group of pictures header. MPEG-1 defines "Closed GOPs" as follows:

Closed GOP

A one bit flag follows the time code. It denotes whether the group of pictures is open or closed. Closed groups can be decoded without using decoded pictures of the previous group for motion compensation, whereas open groups require such pictures to be available.

A typical example of a closed group is shown in figure D.18a.

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

B	B	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	13	14

(b) open or closed group

Figure D.18 -- Example groups of pictures in display order

A less typical example of a closed group is shown in figure D.18b. In this example, the B-pictures which precede the first I-picture must use backward motion compensation only, i.e. any motion compensation must be based only on picture number 2 in the group.

If the closed_gop flag is set to 0 then the group is open. The first B-pictures that precede the first I-picture in the group may have been encoded using the last picture in the previous group for motion compensation.

EX-1005, at 86; EX-2015, ¶76. The "one bit flag" referred to above is the closed_gop flag. EX-2015, ¶76. As the closed_gop flag is a flag in the group of pictures header, it is a syntax element that applies to an entire GOP and not any one specific coded frame. As a result, the closed_gop flag is therefore not an indication of any specific image frame. EX-2015, ¶77.

For example, in Figure D.18 above, the value of the closed_gop flag (0 or 1) defines a property of the entire GOP, such as the 13-picture GOP shown in Fig. D.18a or the 15-picture GOP shown in Fig. D.18b. Unlike the closed_GOP flag, there are other syntax elements in an MPEG-1 video bitstream that pertain to a specific coded picture, namely syntax elements found at the Picture Layer and lower syntax layers. EX-2015, ¶80. For example, MPEG-1 states that "The picture layer contains all the coded information for one picture". EX-1005 at 87. As shown in Figure D.18 above, a closed_gop flag does not indicate any particular image frame. EX-2015, ¶¶79-80.

As Dr. Richardson explains with an example in his Declaration, the presence of an indication in a bitstream necessarily confirms to a decoder the thing indicated. EX-2015, ¶¶112-113. At the time a closed_gop flag is received, a decoder does not have an indication of any image frame—due to transmission errors, there may not *be* a following image frame. EX-2015, ¶114. Any indication occurs later in the bitstream. Indeed, this is the reason for the picture layer in any image frames that follow the closed_gop flag—so that the decoder knows with certainty that when the indication is read, the frame is there. The picture layer (not the closed_gop flag) includes indication of those image frames.

Consider "Temporal Reference" in MPEG-1. Unlike closed_gop flag, Temporal Reference is encoded as a syntax element temporal_reference that is part of the Picture Layer syntax. That is, Temporal Reference is a syntax element that is associated with a specific coded picture. EX-1005, at 38. MPEG-1 discloses that the "Temporal Reference" can be used to define "the order in which the pictures must be displayed." For a given group of pictures (whether open or closed), the first picture in display order has a Temporal Reference of zero:[1]

D.5.3.2 Temporal reference

The Temporal Reference is a ten-bit number which can be used to define the order in which the pictures must be displayed. It may be useful since pictures are not transmitted in display order, but rather in the order which the decoder needs to decode them. The first picture, in display order, in each group must have Temporal Reference equal to zero. This is incremented by one for each picture in the group.

Some example groups of pictures with their Temporal Reference numbers are given below:

Example (a) in	I	B	P	B	P								
display order	0	1	2	3	4								
Example (a) in	I	P	B	P	B								
decoding order	0	2	1	4	3								
Example (b) in	B	B	I	B	B	P	B	B	P	B	B	P	
display order	0	1	2	3	4	5	6	7	8	9	10	11	
Example (b) in	I	B	B	P	B	B	P	B	B	P	B	B	
coded order	2	0	1	5	3	4	8	6	7	11	9	10	
Example (c) in	B	I	B	B	B	B	P	B	I	B	B	I	I
display order	0	1	2	3	4	5	6	7	8	9	10	11	12
Example (c) in	I	B	P	B	B	B	B	I	B	I	B	B	I
coded order	1	0	6	2	3	4	5	8	7	11	9	10	12

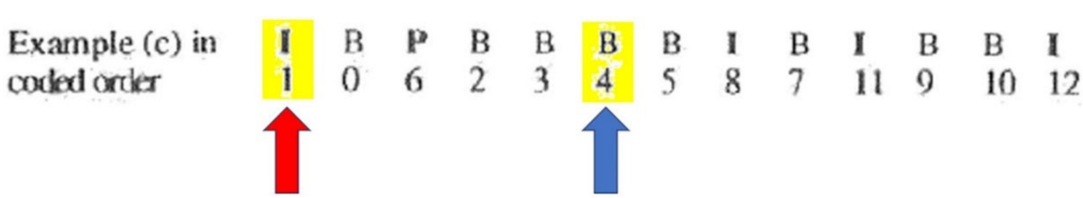
Figure D.19 -- Examples of groups of pictures and temporal references

[1] Temporal reference is also set to zero by the encoder when there are more than 1024 pictures in a GOP. EX-1005, 87. This is a limitation of the number of bits used to express numbers.

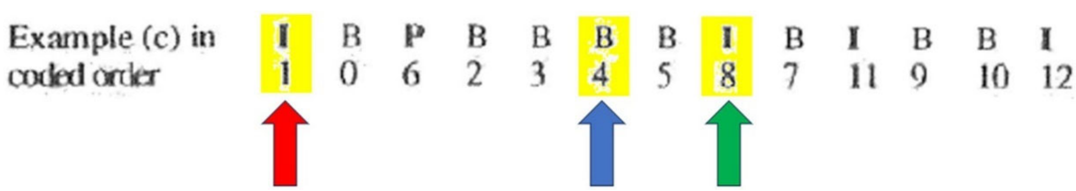
EX-1005, 87-88; EX-2015, ¶¶82-83. Display order is different from the "coded order" of the pictures, which defines the order that the pictures are to be decoded. As illustrated in Figure D.19, the coded order for a picture can be different from the display order.

MPEG-1 suffers from the prior art problems discussed in the '321 Patent because it does not enable random access at a user-specified playback point. MPEG-1 makes clear that after a user specifies a random-access point, the decoder looks for the next sequence start code, group of pictures (GOP) start code, or a picture start code that introduces an I-picture (collectively, "start codes") to start decoding. EX-1005, 43, 87, 120. Then, all prior frames are discarded, and decoding and playback starts from that next start code (rather than from a specific identified frame), which is not the user-specified random-access point. EX-2015, ¶84.

Applying the teachings of MPEG-1 to Example (c) of Figure D.19 highlights the issues with MPEG-1 compared to the claimed inventions of the '321 Patent. As Dr. Richardson explains as illustrated in his Declaration and reproduced below, the first frame of the GOP in decoding order is I_1 (red arrow) and the user selects, for example, frame B_4 (blue arrow) as the playback point. EX-2015, ¶85.

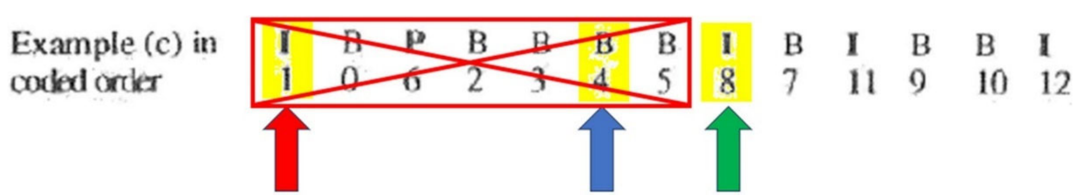


Going on, Dr. Richardson shows that MPEG-1 starts decoding from frame I_8 (green arrow), which is not the first frame I_1 (red arrow) of the independent sequence. MPEG-1 starts decoding from I_8 (green arrow) because MPEG-1 teaches that after a user specifies a random-access point, the decoder looks for the next start code to start decoding. EX-1005, 43, 87, 120. In Example (c), the next start code is a picture start code which introduces an I-picture I_8. *See id.*



EX-2015, ¶87.

Next, Dr. Richardson confirms that the decoder discards all frames until frame I_8 and starts decoding and playback from I_8 because MPEG-1 teaches that all prior frames are discarded and decoding and playback starts from that next start code (rather than from a specific identified frame), which is not the user-specified random-access point. EX-1005, 44, 94, 127.



EX-2015, ¶88.

As another issue, MPEG-1 does not disclose the decoder resetting any identifier value of any indicated first image frame. In the above example, MPEG-1 never teaches resetting any identifier value of those frames. EX-2015, ¶89.

MPEG-1 does not disclose decoding any indication of a first image frame of an independent sequence, starting the decoding from the indicated first image frame, and resetting an identifier value of the indicated first image frame. EX-2015, ¶90. That is, when the user selects a playback point, MPEG-1 discloses that it discards all frames and starts decoding at a start code (not an indicated first image frame). Thus, MPEG-1 suffers from the same random-access problems discussed above, and the ’321 Patent acknowledged MPEG-1’s GOP concept as a problem in random access and sought to address it. ASUS-1001, 2:34-55, 3:52-56; EX-2015, ¶90. For these reasons, MPEG-1 fails to disclose several limitations of the Challenged Claims that enable random access from the user-specified point. EX-2015, ¶90.

2. Kim (U.S. Patent No. 6,912,351)

Unlike the ’321 Patent, Kim is not related to streaming video or random access. Kim is directed to a Time Lapse recording/reproduction device and method. EX-1006, 1:8-12. More specifically, Kim describes recording, compressing and reproducing intermittently recorded image data. In other words, Kim is about single images captured at intermittent time intervals. EX-2015, ¶¶91-92. Examples of where this type of technology is applied is in banks or other security-sensitive areas;

in such implementations, pictures are recorded intermittently and then stored in a database. EX-1006, 1:14-19. These *individual* images can be reproduced when needed. EX-1006, 1:8-56; EX-2015, ¶¶91-92. Unlike streaming video in the ’321 Patent, Kim explains that “only the desired images can be selectively reproduced with high display quality.” EX-1006, 6:17-20. In fact, Kim does not discuss independently decodable video sequences anywhere in its disclosure. This makes sense given Kim’s emphasis on intermittently stored *individual image frames*. EX-1006, 1:38-45.

Kim was concerned that conventional Time Lapse Video Cassette Recorders used magnetic tapes for storage, which resulted in “considerable use and wear” which ultimately deteriorated the quality of the images. EX-1006, 1:57-67. Thus, Kim suggested using an optical disk instead. EX-1006, 2:58-61. Kim’s problem and suggested solution therefore are completely unrelated to the problems and solutions of the ASUS-1001. EX-2015, ¶92.

Petitioner relies heavily on Figure 5A of Kim for nearly every element of the Challenged Claims. *See*, *e.g.*, Pet., 39-51. Figure 5A shows a bitstream that includes both I-frames (images without prediction) and P-frames (images with prediction from previously stored video frames). EX-2015, ¶94. According to Kim’s Figure 5A, each “P-picture frame P1, P2, P3 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." EX-1006, 4:46-52 (shown below).

> The bit stream generated and transmitted to the recording/reproducing unit **140** as described above has a logical structure as shown in, e.g., FIG. **5**A. As shown therein, each P-picture frame **P1**, **P2**, **P3** 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. If image signals are

See EX-2015, ¶94.

But while the IAD *of a P-frame* references the logical address of the corresponding or previous I-frame, the IAD *of an I-frame* is either "zero or absent" because it has no corresponding meaning at all. EX-1006, 4:65-5:7; EX-2015, ¶¶95-99. This is because the I-frame does not rely on prediction from any other frame and therefore it makes no sense for it to have an associated IAD with any meaning. Therefore, as will be discussed later in much more detail, the Petition is incorrect that the IAD of an I-frame can satisfy the claimed "***indication*** of at least one image frame, which is ***the first image frame, in decoding order***, of an independent sequence." *See*, *e.g.*, Pet., 44-51.

Kim's Figure 3 explains the encoding process and how the IAD of the I-frame has no corresponding meaning.

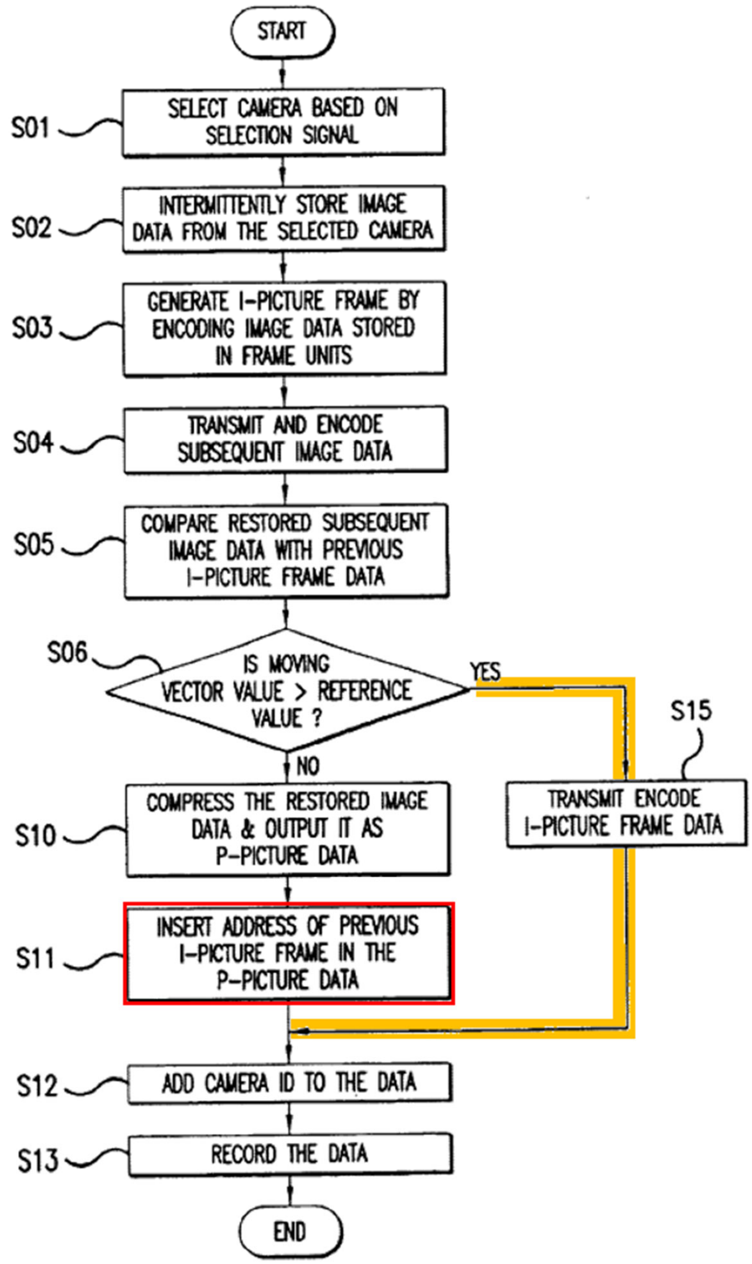


FIG.3

EX-2015, ¶96 (annotating Kim, Fig. 3).

In particular, Kim explains how it encodes the first image as an I-frame, which does not depend on any other frame. EX-1006, 3:57-65 (Step S03). For the next image, Kim teaches that its encoder will determine whether that image frame's data is similar enough to the first image that it can use a P-frame to encode that (second) image frame. EX-1006, 3:65-4:14 (Steps S04-S06). If Kim's encoder determines that the second image frame is similar enough to the first image frame, the encoder will encode the second image frame as a P-frame in Step S10 and then, in Step S11, the encoder will add "the moving vector and the position value of the corresponding (or previous) I-frame [IAD of the P-frame], so that the above information is added to the corresponding compressed P-frame output from the encoder." EX-1006, 4:15-33.

Alternatively, if Kim's encoder determines that the second image frame is *not* similar enough to the first image frame (in Step S06), then the encoder will encode the second image frame as another I-frame (in Step S15). EX-1006, 4:23-27. As shown in the annotated Figure 3 above, the process for encoding the second image frame as an I-frame (shown in orange) skips Step S11 (outlined in red): "insert address of previous I-picture frame in the P-picture data." EX-2015, ¶98. Critically, Kim's encoder therefore only adds "the position value of the corresponding (or previous) I-frame" [IAD] to the *P-frames*, and *not to the I-frames*. EX-1006, 4:15-33. EX-2015, ¶98. For the I-frames, Kim instead teaches that "the address of the corresponding I-picture frame (3) is zero or absent." EX-1006, 4:65-5:1. Thus, the IAD *of the I-frame* does not correspond to a position value or any other meaning.

EX-2015, ¶74. The IAD for the I-frame has no meaning or use, which is why it is allowed to be entirely absent from the bitstream without impact on the method of Kim. EX-2015, ¶74. This is consistent with Kim's claim language, which repeatedly includes "position information" of the corresponding I-frame *for a P-frame* but *not for an I-frame*. EX-1006, claims 1-2, 8, 11. Thus, contrary to the Petition, the IAD for the I-frame cannot satisfy the claimed "indication of at least one image frame, which is the first image frame, in decoding order, of the independent sequence." EX-2015, ¶¶149-199.

Kim's Figure 4 explains the *decoding* process and again confirms that the IAD of the I-frame has no corresponding meaning.

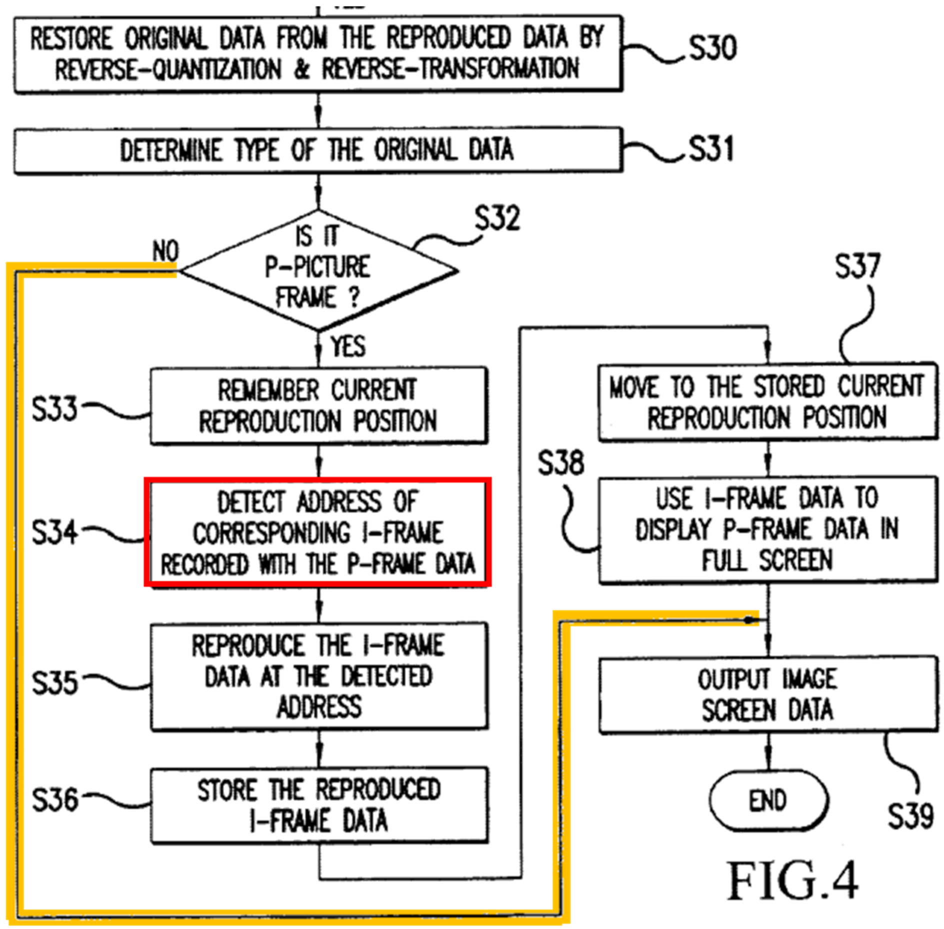


FIG.4

EX-2015, ¶103 (annotating Kim, Fig. 4 (excerpts)).

As shown in the annotated Figure 4 above, Kim explains that at “Step S30, the image data corresponding to the user’s request is transmitted and decoded in the decoder.” EX-1006, 5:29-34. Thus, Kim’s decoder first decodes the image frame that the user selected for decoding, without yet being aware (until Step S31) whether that selected frame is an I-frame or a P-frame. EX-1006, 5:35-38; EX-2015, ¶¶100-104.

If the image frame that the user selected is a P-frame, the corresponding I-frame address (the IAD of the P-frame) is detected at Step S34, and then the corresponding I-frame data is decoded *in part* at Step S36. EX-1006, 5:39-52. The I-frame is not completely decoded at this point because the final reverse-transformation and reverse-quantization steps do not occur until later at Step S38. EX-2015, ¶102. The reverse-transformation and reverse-quantization steps are a required part of the decoding process. EX-1006, 5:32-35. Later, at Step S38, the *partial* P-frame is combined with the reused portions of the *partial* I-frame and the moving (motion) vectors used for prediction, and then the resulting P-frame image is reverse-transformed and reverse-quantized, which completes the decoding process for the full P-frame. EX-1006, 5:55-65. In other words, when the user-requested image to decode is a P-frame, only the P-frame is fully decoded, and the full I-frame is never reverse-transformed and reverse-quantized (only the parts that are used to predict the P-frame). EX-2015, ¶102.

Alternatively, when the image frame that the user selected is an I-frame, the corresponding coded image data is simply read from the optical disk, decoded and displayed. EX-1006, 5:66-6:3. Critically, when the requested image is an I-frame, Step S34 to "detect address of corresponding I-frame recorded with the P-frame data" [IAD] (shown in red in Fig. 4 above) is *skipped* (shown in orange in Fig. 4 above). EX-2015, ¶103. Both the Petition and Dr. Saber show the exact same understanding in their own annotated Fig. 4, *i.e.*, that the IAD is only ever read and decoded *for the P-frame*, and not for the I-frame:

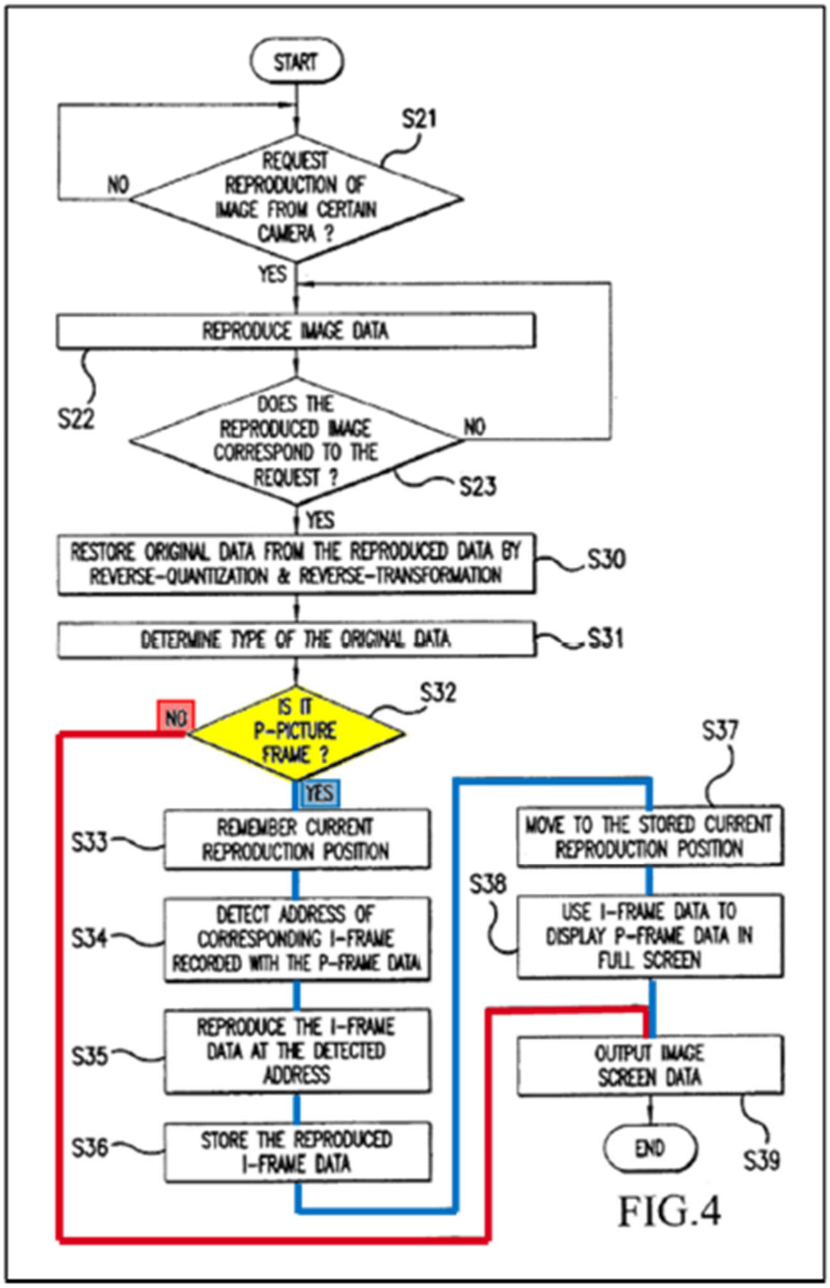


FIG.4

Pet., 50 (colors and annotations in original); EX-1003, ¶55.

Therefore, according to Kim, the IAD is only used for the decoding of a P-frame, while the IAD *of the I-frame* is never read or decoded or used for any purpose, and may not exist, according to Kim. EX-2015, ¶104. A POSITA would have understood that the reason why Kim teaches that the IAD of the I-frame may even be absent from the bitstream entirely is that it makes no difference to the decoder what that value is or whether it exists at all. EX-1006, 4:65-5:1; EX-2015, ¶104. The decoder will never read, decode, or use the value for IAD of the I-frame.

This is consistent with Kim's claim language, which repeatedly includes that the step of "detects the address of the corresponding intra-picture frame" occurs "if the restored image data frame is predictive picture frame data" and not "if the original image frame data type is not predictive picture frame data." EX-1006, claim 11; EX-2015, ¶105.

Therefore, one critical flaw in the Petition that applies to all the Challenged Claims is that Kim's IAD of an I-frame cannot satisfy the claimed "***indication*** of at least one image frame, which is ***the first image frame, in decoding order***, of an independent sequence." *See*, *e.g.*, Pet., 44-50. As explained in further detail below, Kim additionally fails to disclose or render obvious several other limitations of the Challenged Claims of the '321 Patent.

D. Asserted Grounds

Petitioner asserts the following grounds, none of which have a reasonable likelihood of success:

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

Pet., 2.

E. Level of Ordinary Skill

Petitioner alleges that a person of ordinary skill in the art (POSITA) "would have had a (1) Bachelor's degree in electrical engineering, computer engineering, computer science, or a comparable field of study such as physics, and (2) approximately two to three years of practical experience with video encoding/decoding . . . [and] [a]dditional experience can substitute for the level of education, and vice-versa." Pet., 8. Because this Response does not depend on the definition of POSITA, Nokia has applied Petitioner's definition. Nokia, nevertheless, reserves the right to argue in favor of a different level of skill should this Petition be instituted.

F. Claim Construction

Petitioner claims Nokia, in a prior proceeding, construed "indication" in claim 8 to mean "information in an encoded bitstream that necessarily confirms something to a decoder." Pet., 9. Petitioner claims Nokia cited "no support" for this construction in the patent, with the only support coming from Dr. Richardson's report. *Id.* In response, Petitioner effectively argues that something is an "indication" when it merely precedes something else. Pet., 22 ("The GOP header is thus *an indication* of *the first image frame in decoding order* of the GOP because it precedes that image frame.") Petitioner asks the Board to eschew the plain meaning of the term "indicate" for its invented definition without providing any support.

The claims, the specification, and the plain meaning of the term "indication" support Nokia's argument. The patent claims *require* the indication to indicate "at least one image frame," which means the indication must necessarily indicate (or confirm) something. The specification also makes clear that the indication must actually indicate (or confirm) something. *See* ASUS-1001, 4:39-41 ("According to an embodiment, the indication is encoded into the video sequence as a separate flag…"); *see also id.* at 14:39-42 ("With image frame identifiers, the dependencies … are *indicated in an unambiguous manner.*") (emphasis added). This interpretation is confirmed by common sense—an indicator must actually indicate something to qualify as an indicator. The Board should reject Petitioner's proposal, but even if the Board adopts the proposal, the Challenged Claims are not obvious.

Petitioner identifies no claims for construction. Pet., 8-9. Nokia does not believe that any constructions are necessary to deny this Petition. Nokia reserves the right to further address claim construction should this Petition be instituted.

III. ARGUMENT

A. Petitioner Fails to Establish a Reasonable Likelihood That Claims 8, 10, and 11 Are Unpatentable in Light of Ground 1 (MPEG-1).

MPEG-1 does not anticipate or render obvious the asserted claims. First, MPEG-1 does not disclose an indication of the first image frame in decoding order. Instead, MPEG-1 describes a closed_gop flag that indicates whether a group of pictures (GOP) is closed or open. EX-2015, ¶¶107-133. Further, MPEG-1 does not disclose the *decoder* resetting the identifier value for the indicated first image frame ***in decoding order***. Rather, for MPEG-1-compliant bitstreams, the *encoder* sets the temporal reference of the first frame ***in display order*** at the beginning of each new independent sequence (or if there are more than 1024 pictures in a group of pictures). EX-2015, ¶¶139-148.

1. Petitioner fails to demonstrate that MPEG-1 renders obvious claims 8[a], 10[a], and 11[a].

Each of limitations 8[a], 10[a], and 11[a] are written with similar language and require "[decoding/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." Petitioner[2] has failed to show that MPEG-1 teaches or suggests these limitations. EX-2015, ¶107.

Petitioner asserts two distinct arguments for how the "[decoding/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 limitations" are taught or suggested—pointing to MPEG-1's closed GOP or, alternatively, to MPEG-1's Temporal Reference.

a) MPEG-1's Closed_GOP Flag is Not the Claimed "Indication"

Petitioner asserts that MPEG-1's closed_gop teaches or suggests "[decoding/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." This is wrong for at least two reasons: (1) the closed_gop flag is a single-bit flag that only indicates whether the *entire* GOP is closed or open, it does not confirm the presence of any image frame to a decoder. So, closed_gop flag is not an

---

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

indication according to the claims; and (2) not only does closed_gop flag fail to indicate any frame to a decoder, it does not indicate ***which frame is the first***. EX-2015, ¶109. As Dr. Richardson explains in his declaration, the mere fact that closed GOPs happen to have first image frames does not mean that closed_gop flag indicates a first image frame in decoding order to a decoder. EX-2015, ¶110.

MPEG-1's closed GOP flag is only an indication of whether or not the group of pictures is open or closed; the flag does not indicate ***any*** image frame (whether or not the image frame is first in decoding order). EX-1005, 37; EX-2015, ¶110. The closed GOP information at its own structural level of the coded video bitstream data and is not related to any particular frame:

0.5 Structure of the coded video bitstream

This part of ISO/IEC 11172 specifies a syntax for a coded video bitstream. This syntax contains six layers, each of which either supports a signal processing or a system function:

Layers of the syntax	Function
Sequence layer	Random access unit: context
Group of pictures layer	Random access unit: video
Picture layer	Primary coding unit
Slice layer	Resynchronization unit
Macroblock layer	Motion compensation unit
Block layer	DCT unit

EX-1005, 16; EX-2015, ¶110. As Dr. Richardson explains, Syntax elements in the GOP layer apply to an ***entire GOP***, whereas syntax elements in the Picture layer apply to a specific coded picture. This means that when a decoder processes the bitstream and decodes the closed_gop flag, the decoder does not have confirmation about the existence of any particular coded frame in the bitstream. Rather, by decoding closed_gop flag, the decoder instead receives an indication whether a GOP is open or closed based on the value of the flag. EX-2015, ¶111.

There is no confirmation of the existence of any frame (much less the I frame that is required by Petitioner's argument) when the decoder processes the closed_gop flag. EX-2015, ¶111. The decoder remains unaware of any particular frame. For this reason, the closed_gop flag cannot indicate any specific coded frame in an independent sequence, let alone the first one. EX-2015, ¶111. The fact that image frames may follow a closed_gop flag does not mean that the closed_gop flag has served as an indication of whatever frames happen to follow the closed_gop flag. Indeed, other syntax (in the picture layer) serves to indicate any frames to the decoder. Under Petitioner's interpretation of closed_gop flag, a decoder should treat that additional syntax as superfluous, which a POSITA knows is not the case. EX-2015, ¶112.

Put another way, just because a driver gets into their car to go to work does not mean that every time a driver gets into their car, they are going to work. The "getting into the car" cannot be an indication of going to work just because in some instances that occurs.

Further, as Dr. Richardson explains, a POSITA would understand "indication" in the claims refers to information in an encoded bitstream that necessarily confirms something to a decoder. EX-2015, ¶112. Here, the indication is "of at least one image frame, which is the first image frame, in decoding order, of an independent sequence." Thus, as Dr. Richardson confirms, a POSITA would understand that this "indication" cannot merely be information that coincidentally appears alongside other information in the bitstream. EX-2015, ¶112. That is, in video coding, an indication in a bitstream must inform a decoder of something with certainty. EX-2015, ¶112.

As Dr. Richardson explains, an indication takes the following form in a coded bitstream:

> If indication is present, then thing indicated is confirmed necessarily to the decoder.

EX-2015, ¶113. Nonetheless, Petitioner's MPEG-1 Ground relies on a closed_gop flag in MPEG-1 that does not indicate what Petitioner alleges to be the first image frame. EX-2015, ¶114. In MPEG-1 the fact that a closed_gop flag has a value of "0" or "1" 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. Thus, a POSITA would understand that closed_gop flag is not "an indication of at least one image frame, which is the first image frame." EX-2015, ¶114.

Dr. Richardson provides an example of an indication in video coding that a POSITA would understand. The H.264 Standard defines a "NAL unit" as "A syntax structure containing an indication of the type of data to follow and bytes containing that data in the form of an RBSP interspersed as necessary with emulation prevention bytes." EX-2017, ITU-T Rec. H.264 (03/2005) at 8 (as paginated in the document). In H.264, the NAL unit syntax includes a parameter called nal_unit_type that serves as an indication of the type of data contained in a NAL unit. Table 7-1 (excerpted below) provides possible values of nal_unit_type. EX-2017, ITU-T Rec. H.264 (03/2005) at 57 (as paginated in the document):

Table 7-1 – NAL unit type codes

nal_unit_type	Content of NAL unit and RBSP syntax structure	C
0	Unspecified	
1	Coded slice of a non-IDR picture slice_layer_without_partitioning_rbsp( )	2, 3, 4
2	Coded slice data partition A slice_data_partition_a_layer_rbsp( )	2
3	Coded slice data partition B slice_data_partition_b_layer_rbsp( )	3
4	Coded slice data partition C slice_data_partition_c_layer_rbsp( )	4
5	Coded slice of an IDR picture slice_layer_without_partitioning_rbsp( )	2, 3
6	Supplemental enhancement information (SEI) sei_rbsp( )	5
7	Sequence parameter set seq_parameter_set_rbsp( )	0
8	Picture parameter set pic_parameter_set_rbsp( )	1

EX-2015, ¶¶115-117.

As Dr. Richardson explains, A POSITA would understand that, for example, nal_unit_type = 7 serves as an indication of a sequence parameter set. In other words, if nal_unit_type = 7, then the NAL unit is necessarily a sequence parameter set. Consider the opposite: if this were not so, then a decoder would not know with certainty based on nal_unit_type what type of data was contained in the NAL unit. And in that scenario, nal_unit_type would not serve as an indication of the type of data contained in a NAL unit. The decoder would need other information from the bitstream to resolve what type of data was contained in the NAL unit (if the decoder were ever to resolve the uncertainty). EX-2015, ¶117.

So, while the closed_gop flag can indicate whether the GOP is closed, it cannot and does not indicate a first image, in decoding order, of that group of pictures. EX-2015, ¶120.

This distinction is not mere semantics. The "idea of the ['321 Patent's] invention" "is to determine an initiation picture in an independently decodable group of pictures, whereby in the decoding phase any picture prior to said initiation picture is defined as unusable for reference." ASUS-1001, 4:31-38. And in failing to do so, the random access benefits described by the '321 Patent are not achieved. EX-2015, ¶121.

Petitioner cites to Patent Owner's arguments in a prior proceeding and attempts to explain away these arguments. Pet., 22-26 (citing *Amazon.com et al v. Nokia Technologies Oy*, IPR2024-00691, Paper 20 at 32-33 (P.T.A.B. Dec. 19, 2024)). Petitioner is unconvincing.

Petitioner notes that Nokia claims the closed_gop "does not confirm to a decoder any type of image … frame." Pet., 23. Petitioner first avers that this argument only holds if "indication" is defined as "information in an encoded bitstream that necessarily confirms something to a decoder." That is incorrect. Irrespective of how indication is understood, as explained above, closed_gop does not provide any information to the decoder about the particular image frame that follows in the bitstream, much less confirm any type of image as a first image frame.

Petitioner then argues that the closed_gop flag in Petitioner's exemplary GOP "confirm[s] to the decoder that the next image frame decoded will be the first in decoding order." *Id.* Even assuming that a syntax element in the GOP layer indicates/applies to only one individual frame, rather than the entire GOP, as Petitioner argues, this argument is incorrect and unsupported. Petitioner claims that because, in its example GOP, a first image frame happens to follow the closed_gop flag in the bitstream, the closed_gop flag is the one item that necessarily confirms to the decoder that the next image frame will be the first in decoding order. Petitioner provides no support for the assertion that it is the closed_gop flag specifically that confirms that the next frame is first in decoding order, as compared to other information in the GOP header. Thus, even using Petitioner's incorrect understanding of the syntax layers of MPEG-1, where a GOP header that happens to precede a picture in a bitstream (1) indicates a picture at all, (2) indicates that such picture is an I-picture, and (3) indicates that such picture is the first picture in decoding order, Petitioner provides no support for the assertion that the closed_gop flag is the one syntax element of many in the GOP header specifically that signals that the next picture in the bitstream is the first in decoding order.

Perhaps more importantly, Petitioner's "A precedes B, therefore A indicates B" logic is flawed. The closed_gop flag does not always precede an I-frame, which means that it cannot serve the purpose of an indicator. As Dr. Richardson explains, when errors occur in MPEG-1, the first image after the closed_gop flag may not be an I-frame—if there is a transmission error, the first I-frame of a closed GOP may not be transmitted. EX-2015, ¶58, 114. Given this fact, a decoder cannot know whether the frame following the closed_gop flag is an I-frame without reading an actual indication of the frame. Because the closed_gop flag does not confirm with certainty the I-frame, it does not serve as an indicator of the I-frame.

Then, Petitioner argues that because its exemplary closed GOP includes a closed_gop flag before the first decoded image frame, all other examples where the closed_gop flags do not precede the first decoded image frames can be disregarded for obviousness purposes. Pet., 24 fn. 11.

Petitioner misstates Nokia's argument. It is not the case that in some GOPs (namely, Petitioner's "exemplary GOP") the closed_gop flag does serve as the indicator, but others it does not. In no closed_gop—Petitioner's exemplary GOP or otherwise—does the closed_gop flag serve as an "indicator" for the first image frame. Instead, it is the fact that in some GOPs, an I-frame does not follow the closed_gop flag that shows that the flag is not the requisite indicator.

i. Nokia believes the prior Board should not have instituted proceedings

In a prior proceeding on the same ground, the prior Board granted institution, relying on the prior Petitioner's assertion that the following five factors were all true:

(1) a GOP header precedes the first image frame in the GOP in decoding order;

(2) the GOP header contains the closed_gop flag specifying that the GOP is closed;

(3) the closed_gop flag in the GOP header is decoded before the first image frame in the closed GOP is decoded;

(4) the closed_gop flag indicates that the closed GOP is an "independent sequence"; and

(5) the closed_gop flag also indicates that "the next image frame decoded is the first image frame in decoding order" of the "independent sequence," i.e., of the closed GOP.

Amazon.com et al v. Nokia Technologies Oy, IPR2024-00691, Paper 10 (P.T.A.B. Sept. 24, 2024) (hereinafter Amazon Institution Decision). The Board therefore found the "indication of at least one image frame, which is the first image frame, in decoding order, of an independent sequence" is met. *Amazon.com et al v. Nokia Technologies Oy*, IPR2024-00691, Paper 10, 26-27 (P.T.A.B. Sept. 24, 2024). But at least three of these factors are premised on incorrect interpretations of MPEG-1.

First, in item (1), the Board was incorrect that it is always the case that "a GOP header precedes the first image frame in the GOP in decoding order." As Dr. Saber agrees, if playback begins from a random point in the bitstream, an MPEG-1 decoder will "discard all [] bits until it finds a sequence start code, a group of pictures start code, or a picture start code which introduces an I-picture." EX-1003, ¶103.

Thus, the alleged closed_gop flag of MPEG-1 does not disclose the claimed indication of the first image frame in decoding order because the flag only signals to the decoder whether the *entire* GOP is open or closed and does not indicate a first image frame in decoding order, particularly when a user selects for playback a frame in the middle of a video sequence and the closed_gop flag is not decoded. Moreover, as discussed previously, the fact that a first image frame might follow a closed_gop flag does not mean that the closed_gop flag serves to indicate that first image frame to a decoder. In MPEG-1, such an indication is accomplished through picture layer syntax. By this logic, any syntax that exists prior to syntax for a first image frame (e.g., sequence layer and group of pictures layer syntax exists before picture layer syntax in an MPEG-1-compliant bitstream) would serve as an indication of that first image frame. This is inconsistent with how a POSITA would understand an "indication" in a video sequence. EX-2015, ¶¶112-118, 119-123.

For <u>item (3)</u>, "the closed_gop flag in the GOP header is decoded before the first image frame in the closed GOP is decoded," the prior Board and Petitioner assumed that the closed_gop flag is always decoded before the first image frame in a closed GOP is decoded. Amazon Institution Decision at 27. That is incorrect. The closed_gop flag in the GOP header is not always decoded before a first image frame is decoded. For example, when a user selects a frame in the middle of a video sequence, MPEG-1 teaches that all bits are discarded until a start code is reached, as discussed above. In Example (c) of Figure D.19 of MPEG-1 (EX-1005, 87), when the user selects, for example, frame B_4, the first frame that is decoded is frame I_8. Neither the closed_gop flag nor any of the GOP headers were decoded or read in the bitstream. Rather, frame I_8 was decoded first. Thus, the prior Board and Petitioner were incorrect that item (3) is true. Although item (3) may be true in some circumstances, that is not the teaching of MPEG-1. Thus, MPEG-1 does not teach a decoder that decodes an indication of an image frame, which is the first image frame, in decoding order, of an independent sequence. Moreover, as explained, the fact that the closed_gop flag in the GOP header is sometimes decoded before a first image frame is decoded does not mean that the closed_gop flag serves as an indication of that first image frame. By this logic, any syntax decoded before a first image frame would serve as an indication of that first image frame. This is inconsistent with how a POSITA would understand an "indication" in a video sequence. EX-2015, ¶¶112-118, 130.

For <u>item (5)</u>, "the closed_gop flag also indicates that 'the next image frame decoded is the first image frame in decoding order' of the 'independent sequence,' i.e., of the closed GOP," the prior Board and Petitioner made two mistakes. First, the Board and Petitioner conflated correlation with the confirmation required by a

POSITA's understanding of "indication." That is, the Board and Petitioner treated any frame—which was yet to be received or confirmed to a decoder—following a closed_gop flag as having been indicated by that closed_gop flag. But this is incorrect. As Dr. Richardson explains, a decoder that receives a closed_gop flag has no confirmation or indication of any possible subsequent image frame. This is important, because transmission errors such as including interruptions, bit errors, packet losses, DVD/video-CD read errors, truncated files, etc. can mean that the subsequent image frame is not an I-frame. Indeed, the lack of any confirmation on the part of the decoder is the reason for MPEG-1's picture layer syntax that indicates image frames. MPEG-1's picture layer syntax is not superfluous, as a POSITA would know. Second, the prior Board and Petitioner again assumed that the closed_gop flag is always decoded before the first image frame when a closed GOP is decoded. Amazon Institution Decision, at 27. This is incorrect. As discussed above, the first image frame decoded is not necessarily the image frame following the closed_gop flag. Instead, MPEG-1 teaches that the first decoded image frame is the frame following one of the indicated start codes. Given that decoding of start codes and decoding of closed_gop flags are not mutually coextensive, the closed_gop flag is not always decoded and does not indicate the first image frame in decoding order, as discussed above. EX-2015, ¶131. Thus, for similar reasons discussed for items (1) and (3), item (5) is not taught by MPEG-1.

b) MPEG-1's Temporal Reference Is Not the Claimed "Indication"

Alternatively, Petitioner points to MPEG-1's Temporal Reference of zero as teaching "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." Pet., 24-26. A Temporal Reference of zero does not teach this claim element. MPEG-1 does not teach that a Temporal Reference of zero is an indication of the first image frame in ***decoding order*** of ***an independent sequence***. EX-2015, ¶132. Instead, MPEG-1 teaches that "[t]he first picture, in ***display order***, in each group must have Temporal Reference equal to zero." EX-1005, 87; EX-2015, ¶132. MPEG-1 goes on, saying "[t]he Temporal Reference is a ten-bit number which can be used to define the order in which the pictures must be displayed." *Id.* EX-2015, ¶133. As Dr. Richardson explains, the Temporal Reference is not used only for *closed GOPs* (*independent sequences*) but also *open GOPs* (*dependent sequences*). Thus, the Temporal Reference cannot indicate the first image frame in *decoding* order of an *independent* sequence; it is merely the first picture in display order as taught by MPEG-1. EX-2015, ¶133.

D.5.3.2 Temporal reference

The Temporal Reference is a ten-bit number which can be used to define the order in which the pictures must be displayed. It may be useful since pictures are not transmitted in display order, but rather in the order which the decoder needs to decode them. The first picture, in display order, in each group must have Temporal Reference equal to zero. This is incremented by one for each picture in the group.

Some example groups of pictures with their Temporal Reference numbers are given below:

Example (a) in	I	B	P	B	P								
display order	0	1	2	3	4								
Example (a) in	I	P	B	P	B								
decoding order	0	2	1	4	3								
Example (b) in	B	B	I	B	B	P	B	B	P	B	B	P	
display order	0	1	2	3	4	5	6	7	8	9	10	11	
Example (b) in	I	B	B	P	B	B	P	B	B	P	B	B	
coded order	2	0	1	5	3	4	8	6	7	11	9	10	
Example (c) in	B	I	B	B	B	B	P	B	I	B	B	I	I
display order	0	1	2	3	4	5	6	7	8	9	10	11	12
Example (c) in	I	B	P	B	B	B	B	I	B	I	B	B	I
coded order	1	0	6	2	3	4	5	8	7	11	9	10	12

Figure D.19 -- Examples of groups of pictures and temporal references

EX-1005, 87. EX-2015, ¶133.

MPEG-1 specifies a decoding process. MPEG-1 decoders cannot pick and choose parts of MPEG-1 to follow, particularly because MPEG-1 decoders do not have foreknowledge of the bitstreams they will decode. Under MPEG-1's full teaching, it is not true that *any* GOP will always have a first picture in decoding order that is an "I-frame with a Temporal Reference of zero" as the Petition suggests. EX-2015, ¶134. Neither closed_GOP flag nor Temporal Reference indicate any such arrangement. EX-2015, ¶134. For example, Figure D.19 shows many instances where an I-picture is not the first image in display order and an I-picture does not have a Temporal Reference of zero:

D.5.3.2 Temporal reference

The Temporal Reference is a ten-bit number which can be used to define the order in which the pictures must be displayed. It may be useful since pictures are not transmitted in display order, but rather in the order which the decoder needs to decode them. The first picture, in display order, in each group must have Temporal Reference equal to zero. This is incremented by one for each picture in the group.

Some example groups of pictures with their Temporal Reference numbers are given below:

Example (a) in	I	B	P	B	P								
display order	0	1	2	3	4								
Example (a) in	I	P	B	P	B								
decoding order	0	2	1	4	3								
Example (b) in	B	B	I	B	B	P	B	B	P	B	B	P	
display order	0	1	2	3	4	5	6	7	8	9	10	11	
Example (b) in	I	B	B	P	B	B	P	B	B	P	B	B	
coded order	2	0	1	5	3	4	8	6	7	11	9	10	
Example (c) in	B	I	B	B	B	B	P	B	I	B	B	I	I
display order	0	1	2	3	4	5	6	7	8	9	10	11	12
Example (c) in	I	B	P	B	B	B	B	I	B	I	B	B	I
coded order	1	0	6	2	3	4	5	8	7	11	9	10	12

Figure D.19 -- Examples of groups of pictures and temporal references

EX-1005, 87; EX-2015, ¶134.

Thus, Petitioner has failed to show that its partially constructed MPEG-1 decoder discloses an indication of an image frame, which is the first image frame, in decoding order, of an independent sequence as required by the [a] limitations. EX-2015, ¶138.

Petitioner further argues that because in its exemplary GOP, an I-picture *happens to be* the first image frame in decoding order, that exemplary GOP *teaches* an indicator of the first frame in decoding order. Pet., 25. As described above, this happenstance is unintentional, given that MPEG-1 was purpose-built such that B-frames can precede I-frames within a GOP. Temporal Reference is not an indicator of the first frame in decoding order.

2. Petitioner fails to demonstrate that limitations 8[d], 10[d], and 11[d] are obvious over MPEG-1

Each of limitations 8[d], 10[d], and 11[d] are written with similar language and require "[resetting/reset] the identifier value for the indicated first image frame of the independent sequence." Petitioner has failed to show that MPEG-1 teaches or suggests these [d] limitations. EX-2015, ¶139.

Petitioner claims that the Temporal Reference discloses "resetting the identifier value for the indicated first image frame of the independent sequence," but Petitioner fails to establish that the *decoder* resets the identifier value as the claim requires. EX-2015, ¶141.

The claims require the ***decoder*** (not the encoder) to reset the identifier value for the indicated first image frame of the independent sequence. The prior Board, in the Amazon Institution Decision, acknowledged that Temporal Reference is "assigned (in sequence) to the . . . pictures in display order," beginning with 0 for the first picture in display order. Amazon Institution Decision, at 35 (citing EX-1005, 44, 87). Temporal Reference is assigned by the encoder with no further action by a decoder. EX-2015, ¶141. In sum, MPEG-1's temporal reference is only set by the ***encoder*** for the first image frame in display order with no action by a decoder corresponding to the challenged claims. EX-1005, 44, 87. In a prior ITC proceeding, the ALJ concluded the same, holding MPEG-1 did not disclose a decoder performing the required resetting. *See* EX-2018, 52-54. Thus, MPEG-1 fails to teach this claim element. EX-2015, ¶141.

The Temporal Reference is a syntax element whose value is determined by the encoder that informs the decoder which picture is the first in display order "since pictures are not transmitted in display order, but rather in the order which the decoder needs to decode them." EX-1005, 87; EX-2015, ¶142.

It is clear that MPEG-1 does not teach that the ***decoder resets*** the Temporal Reference to 0. As Dr. Richardson explains, at most, it teaches that the ***encoder*** assigns the first picture in display order to zero, and the decoder reads that zero from the bitstream. EX-1005, 87; EX-2015, ¶143. The closest Petitioner comes to showing a "reset" is in arguing that after decoding an exemplary closed GOP, "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." Pet., 29-30. This is not what the claims describe.

The prior Board relied on the same passage. Amazon Institution Decision, at 35-36 (citing to EX-1005, 44). This disclosure and argument fails for two reasons. First, the encoder, not the decoder assigns the Temporal Reference value, and therefore there is no system or method for decoding in MPEG-1 that resets the Temporal Reference. EX-1005, 87; EX-2015, ¶144.

Second, the claim language states "resetting the identifier value for ***the*** first image frame of ***the*** independent sequence." *See* ASUS-1001, Claim 8 (and similar for other challenged claims). That is, the resetting of the identifier value is reset for the independent sequence of the prior limitations. EX-2015, ¶146. This is the antecedent basis used in the claims.

But Petitioner relies (and the prior Board relied) on the Temporal Reference being set to zero for the ***subsequent first image frame of the subsequent*** independent sequence, not the first image frame of the independent sequence being decoded: "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***." Pet., 29-30 (emphasis added). Thus, even putting aside the fact that an MPEG-1 decoder does not reset Temporal Reference, at best Petitioner has pointed to a setting of "the next independent sequence" by the encoder not "the independent sequence" referred to earlier in the same claim, though Petitioner has not actually shown the decoder resetting any identifier value. EX-2015, ¶147.

Thus, Petitioner has failed to show that MPEG-1 decoder discloses resetting the identifier value for the indicated first image frame of the independent sequence. EX-2015, ¶148.

Petitioner again responds to Nokia's arguments in a prior proceeding. Pet., 30-31 (discussing *Amazon.com et al v. Nokia Technologies Oy*, IPR2024-00691, Paper 20 at 50-53 (P.T.A.B. Dec. 19, 2024)). Petitioner restates its argument, claiming that because "the decoder receives and decodes the value of a Temporal Reference … and cannot change the value … [the] decoder resets the Temporal Reference." *Id.* at 31. As discussed above, the encoder sets the value of the Temporal Reference, which the decoder decodes. The decoder does not "reset" anything in decoding a Temporal Reference value that is zero, it merely decodes a value that was already zero. Petitioner acknowledges the passive role of the decoder here, stating "if a decoder unilaterally changed a Temporal Reference value, the decoder would not be able to correctly determine display order." *Id.* Here, Petitioner acknowledges that the decoder is not doing anything beyond decoding—it is not "resetting" anything.

B. Petitioner Fails to Establish a Reasonable Likelihood That Claims 8, 10, and 11 Are Unpatentable in Light of Ground 2 (Kim).

1. Petitioner fails to demonstrate that Kim renders obvious claims 8[a], 10[a], and 11[a].

Each of the Challenged Claims requires "decoding[/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," which is identified as 8[a], 10[a], and 11[a] in the Petition. Pet., Listing of Claims. Petitioner relies solely on Kim's IAD *of the I-frame* as the claimed indication. Pet., 44-51. Again, there are multiple reasons why the IAD of the I-frame cannot satisfy the claimed limitation.

a) Kim's IAD of the I-frame does not indicate anything.

First, the IAD of the I-frame does not disclose or render obvious the claimed "indication" because the IAD of the I-frame is not "an ***indication***" of anything, let alone the required "at least one image frame, which is the first image frame, in decoding order, of the independent sequence." As Kim explains, the IAD of the I-frame may be "zero *or absent*" from the bitstream. EX-1006, 4:65-5:1 (emphasis added); Pet., 47. The Petition specifically relies on situations where the IAD of the I-frame has a zero value (0000). Pet., 46-48. But Dr. Richardson explains that the reason why Kim allows the IAD of the I-frame to have the value of zero *or absent* is because, unlike the IAD *of the P-frame*, the IAD *of the I-frame* has no corresponding meaning and will never be used for anything. EX-2015, ¶154.

Kim describes that the IAD of the P-frame is a "position value" of the corresponding I-frame. EX-1006, 4:27-33, 4:48-52; claims 1-2, 8, 11. But Kim never describes the IAD *of the I-frame* as having any meaning at all. EX-2015, ¶154. This is confirmed by Kim's description of the decoding process, as shown in the annotated excerpt of Figure 4 below:

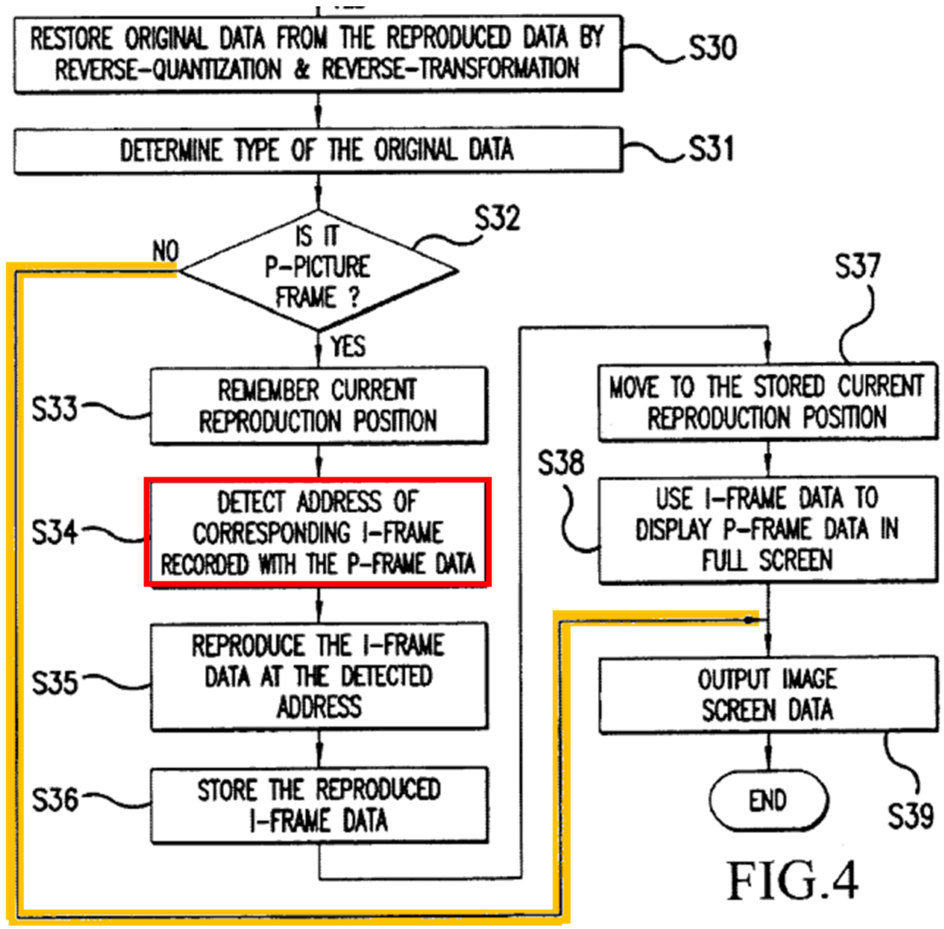


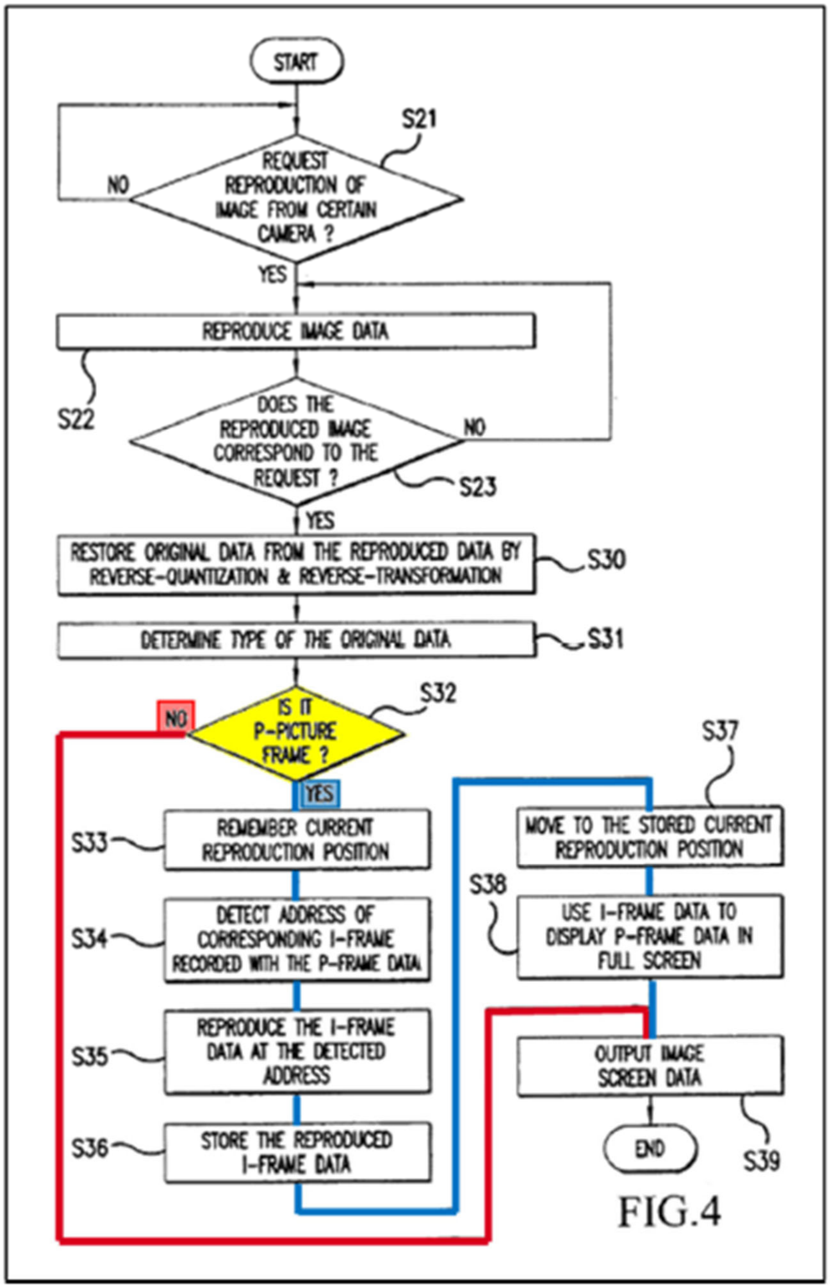
FIG.4

EX-2015, ¶155 (annotating EX-1006, Fig. 4).

As shown in Figure 4, Kim's decoder processes I-frames differently than P-frames. While the process for decoding a P-frame includes Step S34 (outlined in red above) to "detect address of corresponding I-frame recorded with the P-frame data," the process for decoding an I-frame (shown in orange above) *skips* that step. Kim explains that when the user selected image frame is an I-frame, the corresponding coded image data is simply read from the optical disk, decoded and displayed. EX-

1006, 5:66-6:3. Therefore, Kim's decoder never detects the IAD *of the I-frame*. It only detects the IAD *of the P-frame*. EX-2015, ¶156.

Notably, Petitioner agrees that Step S34 of the decoding process to read the IAD only happens for the P-frame. Pet., 42. As the Petition highlights in red below, when the image is an I-frame, Kim's decoder skips Step S34 and does not detect the IAD of the I-frame. Step S34 to detect the IAD, only occurs when the image is a P-frame, as shown in blue below.



Pet., 43 (colors and annotations in original).

Kim never describes any step in which the IAD *of the I-frame* is ever decoded, read, or used for any purpose. A POSITA would have understood that is the reason Kim teaches that the IAD of the I-frame may even be absent from the bitstream entirely because the decoder never uses it. EX-1006, 4:65-5:1; EX-2015, ¶157. This is consistent with the language of Kim's claims, which include the step of "detects the address of the corresponding intra-picture frame" only "if the restored image data frame is predictive picture frame data" (P-frame) and not "if the original image frame data type is not predictive picture frame data" (I-frame). EX-1006, claim 11.

Kim's Figure 6 labeling the IAD of the I-frame as "corresponding I-picture address" does not indicate to a POSITA that there is any such meaning to that value. EX-2015, ¶161. Petitioner admits that the address of the I-frame is its *CAD* value, not its IAD value. Pet., 42-43. The IAD of the I-frame cannot be an address because all of the I-frames have the same "zero or absent" value, and they all clearly cannot be stored at the same location. EX-2015, ¶161.

This is not a situation where the IAD of the I-frame indicates something when it has a 0000 value and indicates nothing when it is absent. The IAD of the I-frame never has any meaning, whether it is zero or absent, which is why it is allowed to be entirely absent from the bitstream in the first place. EX-2015, ¶163. Thus, Kim never discloses or renders obvious that the IAD of the I-frame is the claimed "indication" of anything, even when it takes a 0000 value.

In the previous ITC Investigation that considered Kim and found claims 8 and 11 of the '321 Patent valid over Kim, Dr. Omid Kia testified that "[f]or an I-frame, that IAD does not serve any purpose" and "that value serves no function in the decoding." EX-2019 (1380 Public Tr.), 1733:1-7.

Neither the Petition nor Dr. Saber address this critical flaw in Kim. Nor does the Petition or Dr. Saber assert that there would be any motivation to modify Kim in any way to make the IAD of the I-frame have a corresponding meaning or make it an "indication" as required by the claims. For this reason alone, Kim cannot disclose or render obvious any of the decoding claims 8-11 of the '321 Patent.

b) Kim's IAD of the I-frame does not indicate the required "first image frame, in decoding order."

A second reason Kim does not disclose or render obvious the decoding claims is that the IAD of the I-frame does not indicate the "first image frame, in decoding order" as required. As explained above, the IAD of the I-frame is not an indication of anything. Additionally, Kim's disclosures show that the IAD of the I-frame is certainly not an indication of the first image in decoding order.

Kim explains that the first image in decoding order is whichever image the user selects to decode. EX-1006, 5:29-34 ("In Step S30, the image data corresponding to the user's request is transmitted and decoded in the decoder").

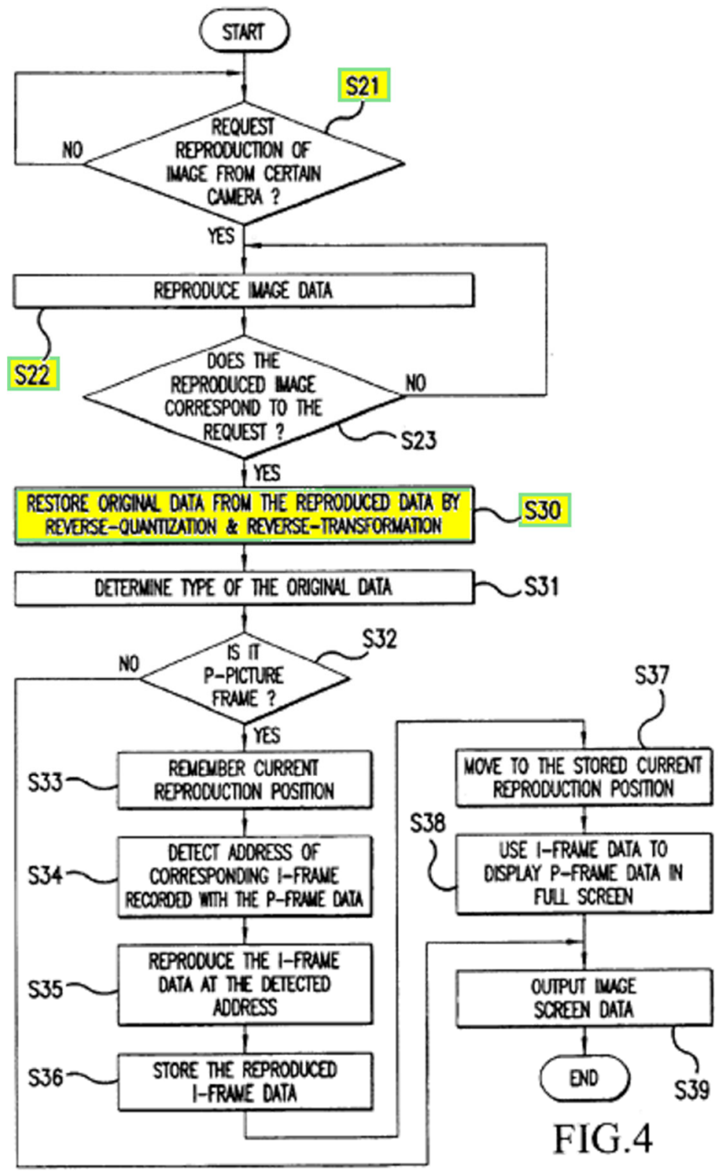


FIG.4

EX-1006, Fig. 4 (annotations added to Steps S20, S21, S30).

There is no consideration of the value of the IAD of the I-frame to indicate to the decoder what frame should be the first image frame in decoding order. EX-2015, ¶168. Kim's decoder simply decodes the picture that the user selected in Steps S21 and S22. EX-1006, 5:13-32. Indeed, Kim's decoder decodes the selected image frame in Step S30 before even determining whether the image frame was an I-frame or a P-frame at Step S31. EX-1006, 5:30-38; EX-2015, ¶169.

In the example where a user selects a P-frame to decode, Kim's decoder partially decodes the P-frame (which is only part of the full P-frame because the resulting full P-frame will also include the partial I-frame) in Step S30, including the final decoding steps of reverse-quantization and reverse-transformation (shown in yellow below). EX-1006, 5:30-35. When the partial P-frame's is decoded at Step S30, the IAD of the I-frame has not been read or decoded. EX-1006, Fig. 4; EX-2015, ¶170 (explaining how Step S34 occurs well after Step S30, and only occurs when the image frame is a P-frame). This shows that the IAD of the I-frame does not indicate the first image frame in decoding order. The first image frame in decoding order is whatever image frame the user selects, and has no relation to the value of the IAD for the I-frame. EX-2015, ¶170.

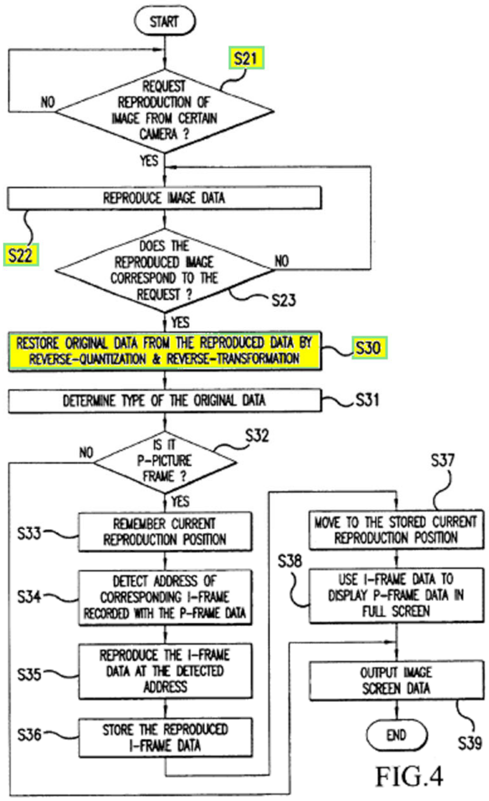


FIG.4

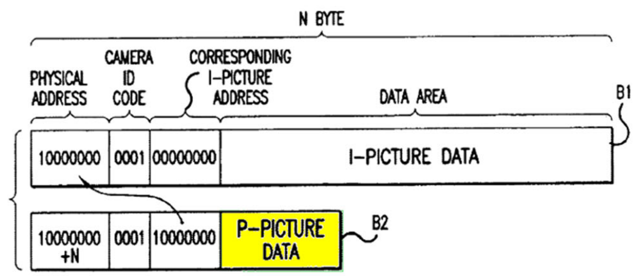


FIG.6

EX-2015, ¶170 (annotating EX-1006, Fig. 4 and Fig. 6).

Petitioner does not address this problem with Kim in the context of claim 8[a]. To the extent that the Petition's discussion for decoding claim 8[b] applies here, Petitioner is wrong that Kim decodes the I-frame before decoding the P-frame even when the user selects a P-frame. Pet., 54-55. Petitioner asserts that "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." *Id*. However, Kim expressly states that the reverse-quantization and reverse-transformation steps of the decoding process do not occur for the I-frame itself, but rather only occur for the parts of the I-frame that are used for prediction of the P-frame *after* those parts are combined with the rest of the P-frame data at Step S38. EX-1006, 5:55-65; EX-2015, ¶171-172. Therefore, even though the P-frame needs portions of the I-frame to complete the full image reproduction (shown in blue below), Kim's decoder does not decode the full I-frame first; rather, when the user selects a P-frame, only the P-frame is fully decoded. EX-2015, ¶171-174.

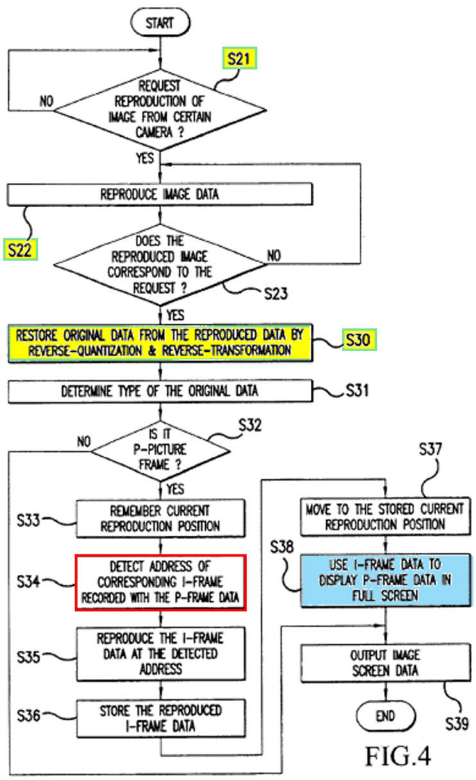


FIG.4

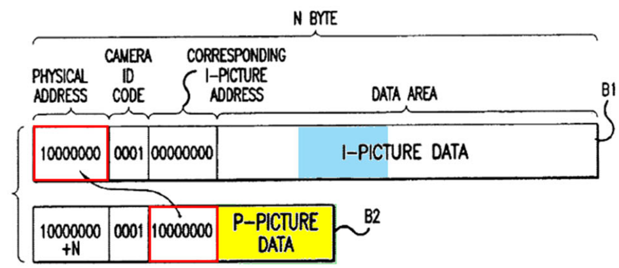
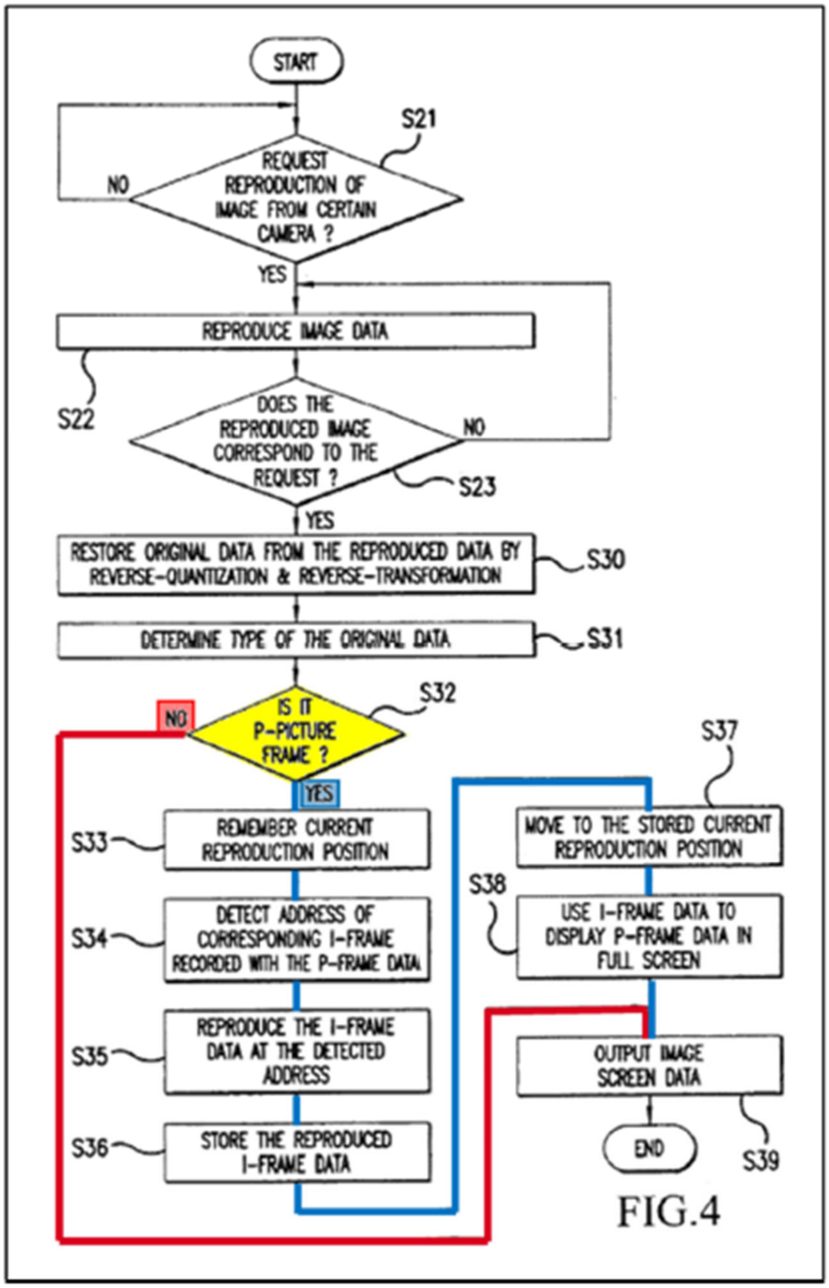


FIG.6

EX-2015, ¶172 (annotating EX-1006, Fig. 4 and Fig. 6).

Rather than analyze the actual disclosures of Kim, the Petition merely concludes without explanation that "[b]ecause 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*).” Pet., 47 (emphasis in original). Again, Petitioner provides no evidence or explanation as to how the IAD of the I-frame *indicates* that the picture is “the first image frame, *in decoding order*, of an independent sequence” as required by the claims.

In fact, the Petition agrees that when the selected image frame is an I-frame, the decoding process in Kim *skips* the step of reading the IAD value. Pet., 42-43.



Pet., 43 (EX-1006, Fig. 4 (annotations added)

As Kim's decoding process explains, Step S34 of Figure 4 to detect the IAD is only performed for a P-frame (blue path in the annotated Figure 4 above), and that step is skipped for an I-frame (red path in the annotated Figure 4 above). EX-1006, 5:66-6:3; EX-2015, ¶¶171-174. The Board also acknowledged this in a prior proceeding. *Element TV Company*, IPR2024-00691, Paper 10, 58 (P.T.A.B. Sept. 24, 2024) ("[a]fter decoding the image data, Kim's controller uses the IAD in the image data, e.g. to 'detect[] the address of the corresponding I-picture frame' ***if the image data is a P-picture***.") (emphasis added).

Thus, no matter what type of image frame is selected for decoding by the user, the IAD *of the I-frame* (the alleged "indication" in the Petition) is never read, decoded, or used for anything in Kim. EX-2015, ¶167-174. The image that is decoded first is the image that is selected by the user; there is no indication of any first frame in decoding order coming from the IAD of the I-frame. EX-2015, ¶¶167-174. In the previous ITC Investigation, Dr. Omid Kia agreed that "[i]t is the user that selects the picture to decode" and not the value of the IAD of the I-frame. EX-2019, 1730:10-1731:2. Therefore, the IAD of the I-frame cannot be "an ***indication*** of at least one image frame, which is ***the first image frame, in decoding order***, of the independent sequence."

That there may, out of happenstance, be an occasion where an I-frame was selected by a user to be decoded first, and the IAD of that I-frame has a zero value, does not transform that IAD value into an indication of anything. It is still unrebutted that there are no steps in the encoding or decoding process of Kim that ever uses the IAD *of the I-frame* to indicate an image frame to decode first. EX-2015, ¶¶167-174.

Neither the Petition nor its expert, Dr. Saber, address this critical flaw in Kim. Nor does the Petition or Dr. Saber assert that there would be any motivation to modify Kim in any way to make the IAD of the I-frame have a corresponding meaning or make it an "indication" as required by the claims. For this additional reason, Kim cannot disclose or render obvious any of the decoding claims 8-11 of the '321 Patent.

Addressing the prior Amazon proceedings, Petitioner asserts that a frame comes first in decoding order when decoding of the frame is finished, not begun. Pet., 49. For support, they point to an MPEG-1 discussion (not a Kim one) that defines the bitstream order as "the order in which the decoder requires them to decode the bitstream." EX-1005, 71. Assuming that bitstream order and decoding order are synonymous and assuming that MPEG-1's definition of bitstream order is also a POSITA's (and thus applies to Kim), Petitioner's argument still fails. "[T]he order in which [the frames] are required" does not specify that a frame is "required" when it is decoded in full, rather than decoded in part, as Petitioner claims. Rather, a frame is "required" not when it is fully decoded, but rather when *any part of the frame* is read or decoded. To claim otherwise would be to claim that a frame that is partially but never fully decoded is not "required," only because it is not fully decoded, even if part of the frame is actually necessary for reproducing another frame. Thus, if Petitioner's read of the MPEG-1 bitstream order definition is correct and applies to Kim, not just MPEG-1, a frame could be necessary to decode another frame but still not "required," so long as it is not decoded in full.

c) Kim's IAD of the I-frame is never "decode[d] from the video sequence."

A third reason Kim does not disclose or render obvious the decoding claims is that the IAD of the I-frame is never decoded, as required by the claims. Petitioner and Dr. Richardson agree that when an I-frame is selected by the user to decode, the decoding process *skips* the Step S34 that detects the IAD. Pet., 42-43; EX-2015, ¶179. The Step S34 to detect the IAD is only performed when a P-frame is selected. Therefore, the IAD *of the I-frame* (the alleged "indication") is never decoded. EX-2015, ¶179.

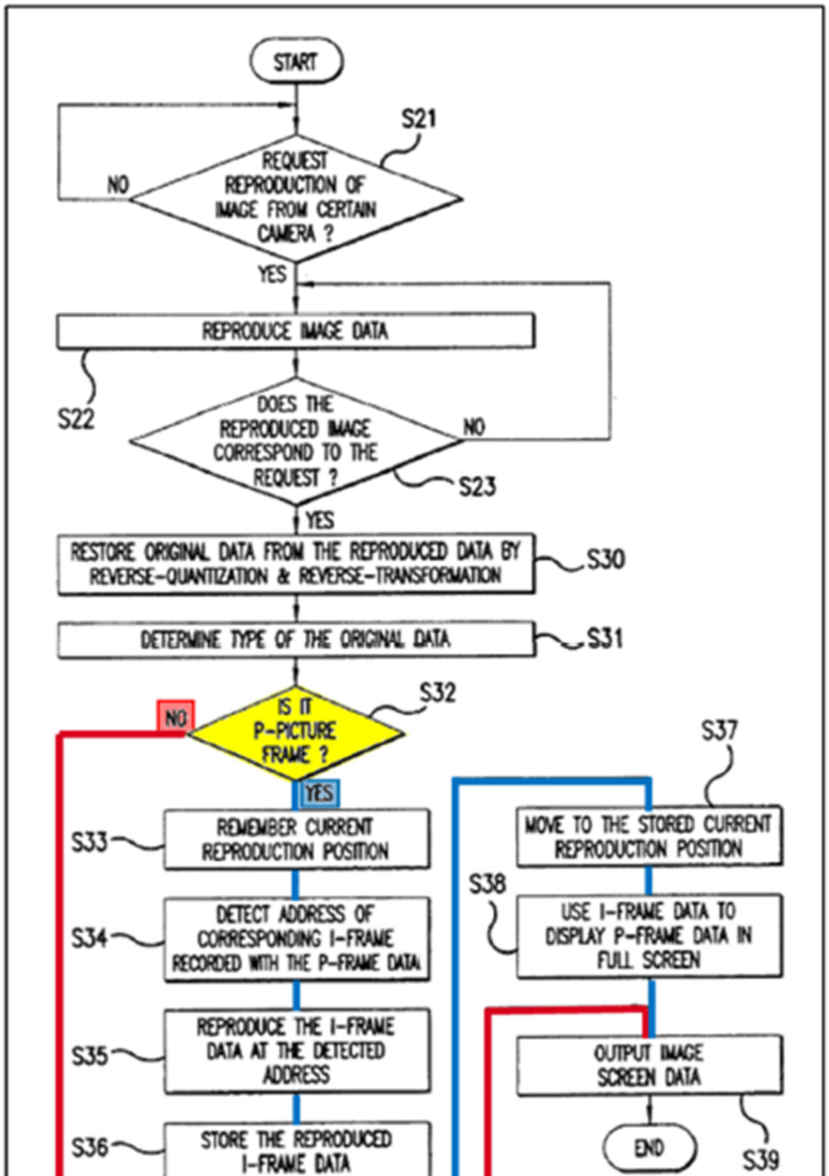


FIG.4

EX-1003, ¶126 (EX-1006, Fig. 4 (annotations added)).

Although Dr. Saber describes and shows in his own annotated Figure 4 that Kim's decoder does not detect the IAD for the I-frame, he nevertheless concludes that a "A POSITA would have [] understood that Kim's decoder decodes the IAD of image frames." EX-1003, ¶132. He cites to Figure 5A to state that "a POSITA would have understood that [decoding the bitstream in Figure 5A] include[s] decoding the components of the bitstream, which would include IAD values…." EX-1003, ¶132.

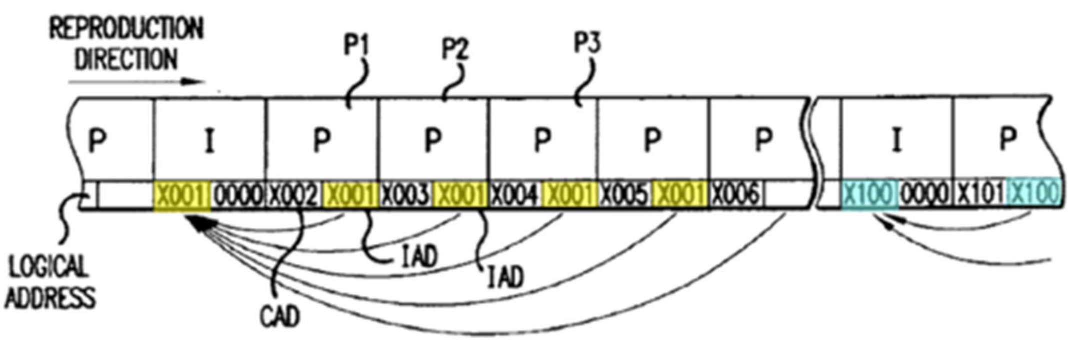


FIG.5A

EX-1003, ¶131 (EX-1006, Fig. 5A (annotations added)).

However, Dr. Saber's opinions are not supported by the figure or the description in Kim. Figure 5A, shown above, does not include any arrows that point to or from the IAD *of the I-frame*. Rather, a POSITA would understand that Figure 5A is consistent with Figure 4 and the description in the specification that the decoding process of Kim never includes decoding the IAD *of the I-frame* but rather only decodes the IAD *of the P-frame* (which points to the *CAD* of the I-frame). EX-2015, ¶180-81.

It is also the case when the user selects a P-frame to decode that the IAD *of the I-frame* is not decoded. The annotated Figure 4 above, and the description in

Kim's specification, show that when the user selects a P-frame, only the IAD *of the P-frame* is decoded, not the IAD *of the I-frame*. EX-1006, 5:66-6:3; EX-2015, ¶179.

Neither the Petition nor Dr. Saber assert that there would be any motivation to modify Kim in any way to include decoding the IAD of the I-frame. For this additional reason, Kim cannot disclose or render obvious any of the decoding claims 8-11 of the '321 Patent.

d) Kim does not have a "video sequence."

A fourth reason Kim does not disclose or render obvious the decoding claims is that Kim does not disclose or render obvious "decoding from the *video sequence*." For the same reasons as explained above for the encoding claims, Kim expressly teaches decoding a single image when it states that "only the desired images can be selectively reproduced with high display quality." EX-1006, 6:17-20. Thus, Kim teaches the select encoding and decoding of individual images at a time, and not the video sequence of the '321 Patent. EX-2015, ¶185.

Again, neither the Petition nor Dr. Saber even attempt to explain how decoding of single images in Kim can satisfy a video sequence of claim 8[a], 10[a], or 11[a]. Pet., 44-51, 66, 69; EX-1003, ¶¶127-137, 159-169. Nor does the Petition or Dr. Saber assert any motivation to modify or combine Kim in such a way as meet the "video sequence" of claim 8[a], 10[a], or 11[a].

Dr. Saber asserts that it would have been obvious to modify Kim to decode a video sequence. EX-1003, ¶¶144. To support this, he points to Figure 4, which he claims contemplates a video sequence merely by the mention of I- and P-frames. *Id.* As explained above, Kim describes reproducing *individual* images from intermittent time lapse recording using I-frame and P-frame terminology. Kim does not contemplate, nor does it suggest, decoding video sequences.

A POSITA, in fact, would not have been motivated to modify Kim's decoding process to repeat itself over and over again for all of the image frames in view of Kim's express teaching to decode only single selected images. EX-1006, 6:17-20; EX-2015, ¶189. A POSITA would have understood that modifying Kim to repeatedly decode images into a video sequence would have required more effort, processing power, memory, and cost. EX-2015, ¶189. Such a scheme would not have been efficient and necessary for the applications that Kim targeted, which only need individual time lapse images rather than a video sequence. EX-1006, 1:14-20; EX-2015, ¶189.

Therefore, a POSITA would not have been motivated to modify Kim to decode a video sequence, as that is the opposite of what Kim teaches.

2. Petitioner fails to demonstrate that Kim discloses or renders obvious claims 8[b], 10[b], and 11[b].

Each of the independent decoding claims requires "***start[ing] the decoding*** of the video sequence ***from said first image frame*** of the independent sequence," which is identified as 8[b], 10[b], and 11[b] in the Petition. Pet., Listing of Challenged Claims.

The key piece of this limitation that is never addressed by either the Petition or Dr. Saber is that "said first image frame" in 8[b] must be "*the* first image frame" that was identified by the "indication" of 8[a], 10[a], and 11[a]. *E.g.*, ASUS-1001, claim 8[a] ("an ***indication*** of at least one image frame, which is ***the first image frame, in decoding order***, of an independent sequence"). As explained above with respect to 8[a], 10[a], and 11[a], the decoding of an image frame in Kim has nothing to do with the IAD of the I-frame but is instead simply the image frame that is selected by the user. *See supra* Section III.B.1.

As explained above, if the user selects an I-frame to decode, Kim's decoder decodes the I-frame without ever reading the IAD of the I-frame and therefore without regard to the image frame being "said image frame" from any indication in the bitstream. EX-1006, 5:13-34, 5:66-6:3, Fig. 4; EX-2015, ¶¶193-197. Likewise, if the user selects a P-frame to decode, Kim's decoder decodes the P-frame without ever reading the IAD of the I-frame and therefore without regard to the image frame being "said image frame" from any indication in the bitstream. EX-1006, 5:13-65, Fig. 4; EX-2015, ¶¶193-197.

Therefore, even if the IAD of the I-frame somehow was the claimed "indication" of claims 8[a], 10[a], and 11[a], Kim's actual decoding process does not start decoding "from said first image frame" because it simply starts decoding from whatever frame that the user selected. EX-2015, ¶¶193-197; EX-2019, 1730:10-1731:2.

3. Petitioner fails to demonstrate that Kim discloses or renders obvious claims 8[d], 10[d], and 11[d].

Each of the independent decoding claims requires "resetting[/reset] the identifier value for the indicated first image frame of the independent sequence," which is identified as 8[d], 10[d], and 11[d] in the Petition. Pet., Listing of Challenged Claims. Petitioner asserts that the IAD of the I-frame is reset to zero for each I-frame in Figure 5A of Kim. Pet., 60.

As a preliminary matter, the decoder, not the encoder, must reset the identifier value. In the previous ITC Investigation, the experts on both the Complainant side and Respondent side agreed that the resetting steps in claims 8 and 11 had to be performed by the decoder. EX-2019 (1380 Public Tr.), 601:8-10 ("Q: According to the claims, what has to perform the resetting step? A: The decoder."), 1065:12-1066:3 ("Q: It would not meet the claim limitation to just read a value that was reset by an encoder, because the claim requires that the resetting be done by the decoder, right, sir? A: That's correct."), 1732:15-18. There is simply no serious dispute that elements 8[d], 10[d], and 11[d] must be performed by the *decoder* and not the *encoder*. *See* ASUS-1001, 11:26-30.

That distinction is dispositive. Kim fails to disclose or render obvious that the IAD of the I-frame is ever reset by the *decoder*. In fact, Kim's Figure 5A, on which Petitioner relies, shows "logical data structure of a digital data stream *compressed and recorded* on the optical disk" by Kim's *encoder*, not its decoder. EX-1006, 2:49-51 (emphasis added). Kim makes clear that the IAD values are generated by and then received from the *encoder*. EX-1006, 4:34-45 ("the address of the I-picture data received from the encoder 110"), Fig. 3.

Petitioner makes only one argument that it is the decoder, and not the encoder, in Kim that resets the IAD of the I-frame. Petitioner asserts that "[w]hen Kim's decoder 'reproduces' the bitstream, it resets the IAD…" Pet., 63-64. In essence, Petitioner's only argument is that because the decoder passively reproduces the work of the encoder, it somehow carries out the act of "resetting." This argument does not satisfy the "resetting" requirement of 8[d], 10[d], and 11[d].

Indeed, in the previous ITC Investigation, the expert for the patent challenger, Dr. Eli Saber (who is also the expert for the Petitioner in this proceeding), testified that it "would not meet the claim limitation to just read a value that was reset by an encoder, because the claim requires that the resetting be done by the decoder." EX-2019, 1065:19-1066:3. Dr. Saber then presented the same arguments as this Petition—and makes the same arguments in his expert report—regarding how Fig. 5A shows the IAD value of the I-frame resetting. EX-2019, 1038:3-18. And, in response, Dr. Omid Kia explained, just as Dr. Richardson does now, that Kim's Figure 5A and other teachings are related to the *encoder*, in contrast to the claims, which require the *decoder* perform the resetting. EX-2019, 1732:11-1733:9; EX-2015, ¶¶193-197.

And when considering the same argument advanced here "that the decoder reading a 'reset' value from a bitstream and storing it locally meets the resetting limitation," the ALJ correctly concluded "[t]here is really no dispute that . . . the alleged 'identifier value' in Kim is set/reset by the encoder and the decoder simply reads or derives it from the bitstream," and therefore "Kim does not teach a decoder resetting the IAD value." EX-2018, 56-57 (also rejecting that Kim rendered obvious the '321 Patent).

The reason why, in the '321 Patent, the decoder resets the identifier is so that the decoder can define a point to start decoding the video when a user selects a random access point. ASUS-1001, 11:26-40 ("[T]his [random access problem] 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 to zero .... Consequently, when activated at a random point, the ***decoder preferably sets*** the identifier of the first image frame, preferably an I-frame, of the independently decodable sub-sequence.") (emphasis added); EX-2015, ¶¶193-97. Kim has no reason to teach or suggest the solution of the ’321 Patent because Kim has nothing to do with random access during streaming video. Kim’s single image reproduction for a time lapse recording on an optical disk is not related to the same problem or solution of the ’321 Patent.

Therefore, the Board should conclude just as the ALJ did in the previous ITC Investigation that Kim does not teach or render obvious the claims of the ’321 Patent for all of the reasons explained above.

C. Petitioner Fails to Establish a Reasonable Likelihood That Claim 9 Is Unpatentable in Light of Ground 3 (MPEG-1 in combination with Yagasaki).

Petitioner cannot show it met its burden for Ground 3. Claim 9 depends on Claim 8 and Petitioner relies on its Ground 1 to show the predicate independent claim is taught or suggested. But, as shown above in III.A.1-2, MPEG-1 fails to teach or suggest at least limitation 8[a] and 8[d]. EX-2015, ¶¶107, 139. For that reason, Ground 3 fails.

IV. CONCLUSION

For any or all of the above described reasons, Nokia respectfully requests that the Board deny institution.

Dated: October 29, 2025

Respectfully submitted,

/s/ Scott W. Hejny
Scott W. Hejny
Reg. No. 45,882
shejny@mckoolsmith.com
Eric S. Hansen
Reg. No. 63,095
ehansen@mckoolsmith.com
MCKOOL SMITH, P.C.
300 Crescent Court, Suite 1200
Dallas, TX 75201
Telephone: 214.978.4000
Facsimile: 214.978.4044

Counsel for Patent Owner,
Nokia Technologies Oy

CERTIFICATION OF WORD COUNT

The undersigned hereby certifies that the portions of the above-captioned **PATENT OWNER'S PRELIMINARY RESPONSE** specified in 37 C.F.R. §42.24 have13,953 words, in compliance with the 14,000 word limit set forth in 37 C.F.R. §42.24(b)(1). This word count was prepared using Microsoft Word 2010.

Dated: October 29, 2025

By: */s/ Scott W. Hejny*
Scott W. Hejny
Reg. No. 45,882

CERTIFICATION OF SERVICE

The undersigned hereby certifies that a copy of this **PATENT OWNER'S PRELIMINARY RESPONSE PURSUANT TO 37 C.F.R. § 42.10(b)** has been served on Petitioner via email on the following counsel of record for Petitioner:

Jeremy J. Monaldo
IPR54587-0015IP1@fr.com

Kim Leung
Joseph V. Colaianni
Linhong Zhang
Jack R. Wilson IV

Dated: October 29, 2025

By: */s/ Scott W. Hejny*
Scott W. Hejny
Reg. No. 45,882