

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

LG Electronics, Inc.,

Petitioner

v.

Broadcom Corporation,

Patent Owner

Patent No. 7,342,967

Filing Date: December 11, 2002

Issue Date: March 11, 2008

Title: SYSTEM AND METHOD FOR ENHANCING PERFORMANCE OF
PERSONAL VIDEO RECORDING (PVR) FUNCTIONS ON HITS DIGITAL
VIDEO STREAMS

Case No. IPR2017-01544

PETITION FOR *INTER PARTES* REVIEW

TABLE OF CONTENTS

I.	PRELIMINARY STATEMENT	1
II.	THE '967 PATENT	2
A.	Background	2
B.	Challenged Claims	7
1.	Claim 1	7
2.	Claim 2	8
3.	Claim 3	8
4.	Claim 4	8
5.	Claim 5	9
C.	Relation to CIP Parent Application	9
III.	LEVEL OF ORDINARY SKILL	11
IV.	CLAIM CONSTRUCTION	11
A.	“instructions” (claim 1)	12
B.	“without having decoding the at least one slice below the intracoded slice” (claim 1)	14
C.	“transport packet” (claim 1)	14
V.	CLAIMS 1-5 OF THE '967 PATENT ARE UNPATENTABLE OVER THE PRIOR ART	15
A.	Ground 1: MacInnis anticipates claims 1-5.	15
1.	Claim 1	23
2.	Claim 2	38

3.	Claim 3: MacInnis discloses “wherein causing decoding the at least one slice above the intracoded slice in the another picture comprises causing: predicting the slices above the intracoded slice in the another picture from the intracoded slice and the slices above the intracoded slice in the picture.”	41
4.	Claim 4: MacInnis discloses “wherein decoding the at least one slice above the intracoded slice further comprises causing: decoding the at least one slice in raster order.”	44
5.	Claim 5: MacInnis discloses “wherein the picture comprises a P-picture from a HITS stream.”	46
B.	Ground 2: The combination of MacInnis and ISO 13818-2 renders claim 4 obvious.....	47
1.	Claim 4: MacInnis in view of ISO 13818-2 renders obvious “wherein decoding the at least one slice above the intracoded slice further comprises causing: decoding the at least one slice in raster order.”	47
C.	Ground 3: Chen ‘677 anticipates claims 1-4.....	50
1.	Claim 1	53
2.	Claim 2	66
3.	Claim 3: Chen ‘677 discloses “wherein causing decoding the at least one slice above the intracoded slice in the another picture comprises causing: predicting the slices above the intracoded slice in the another picture from the intracoded slice and the slices above the intracoded slice in the picture.”	68
4.	Claim 4: Chen ‘677 discloses “wherein decoding the at least one slice above the intracoded slice further comprises causing: decoding the at least one slice in raster order.”	70

D.	Ground 4: Chen '677 renders claim 5 obvious.	72
1.	Claim 5	72
E.	Ground 5: The combination of Chen '677 and ISO 13818-2 renders claims 1-5 obvious.....	74
1.	Claim 1	74
2.	Claim 4: Chen '677 in view of ISO 13818-2 renders obvious “wherein decoding the at least one slice above the intracoded slice further comprises causing: decoding the at least one slice in raster order.”	77
VI.	MANDATORY NOTICES	80
A.	Real Party-in-Interest	80
B.	Related Matters.....	80
C.	Lead and Back-Up Counsel, and Service Information	80
VII.	CERTIFICATION UNDER 37 C.F.R. § 42.24(d)	81
VIII.	GROUND FOR STANDING.....	81
IX.	STATEMENT OF PRECISE RELIEF REQUESTED FOR EACH CLAIM CHALLENGED	81
X.	CONCLUSION	82

LIST OF EXHIBITS

- | | |
|---------------|---|
| Exhibit 1001. | U.S. Patent No. 7,342,967 (“the ’967 patent”). |
| Exhibit 1002. | Declaration of Dr. Edward J. Delp, Ph.D. and Curriculum Vitae. |
| Exhibit 1003. | U.S. Patent Application Publication No. 2002/0061183 to MacInnis (“ <i>MacInnis</i> ”). |
| Exhibit 1004. | U.S. Patent Application Publication No. 2001/0026677 to Chen et al. (“ <i>Chen</i> ’677”). |
| Exhibit 1005. | U.S. Provisional Patent Application 60/239,425 to MacInnis. |
| Exhibit 1006. | U.S. Patent Application 09/951,693 to Demas et al (“CIP parent application”). |
| Exhibit 1007. | U.S. Patent No. 6,940,904 to Chen et al. (“ <i>Chen</i> ’904”). |
| Exhibit 1008. | Exhibit 69 from Complaint in <i>Certain Semiconductor Devices and Consumer Audiovisual Products Containing the Same</i> , ITC-337-TA-1047. |
| Exhibit 1009. | ISO/IEC Standard 13818-2:2000: “Information technology- Generic coding of moving pictures and associated audio information: Video.” (“ISO 13818-2”) |
| Exhibit 1010. | Excerpts from IEEE 100 The Authoritative Dictionary of IEEE Standards Terms, 7 th Edition (2000). |
| Exhibit 1011. | Excerpts from Newton’s Telecom Dictionary 16 th Edition (2000). |
| Exhibit 1012. | Excerpts from Digital Video: An Introduction to MPEG-2, Barry G. Haskell, et al. (1997). |
| Exhibit 1013. | Declaration of Christopher C. Johns |
| Exhibit 1014. | Library of Congress Online Catalog record for Digital Video: An Introduction to MPEG-2 (Exhibit 1012). |

Petitioner LG Electronics, Inc., requests *inter partes* review of claims 1-5 of the '967 patent (EX-1001), currently assigned to Broadcom Corporation.

I. PRELIMINARY STATEMENT

The '967 patent purports to disclose a new way to reduce the processing associated with building pictures during rewind of a HITS digital video stream. EX-1001 2:63-3:10. But the patent admits that it was known how to rewind a HITS digital video stream. *Id.* 2:31-37. And all the patent purports to contribute to the art is not decoding portions of pictures that will not be displayed. *Id.* 2:63-3:10.

The '967 patent claims “decoding at least a portion of another picture...without [] decoding the at least one slice below the intracoded slice.” EX-1001 claim 1. But doing so was not new to the art; it was only new to the application that matured into the '967 patent. The claim feature was not in any parent application to which the '967 patent claims priority, so the claims of the '967 patent are only entitled to the filing date of the application that matured into the '967 patent, making another Broadcom application filed more than a year earlier prior art under 35 U.S.C. § 102(a).

Just like the '967 patent, this other Broadcom application, Publication No. 2002/0061183 to *MacInnis*, discloses how to rewind a HITS digital video stream, EX-1003 ¶¶[0037]-[0038], [0046], [0105]-[0134], recognizing that “not all slices

[in every picture] must be decoded.” *Id.* ¶[0111]. Instead, *only* “necessary slices” are decoded, which can double the average decoding rate. *Id.*

Another prior art reference, Publication No. 2001/0026677 to *Chen*, published more than a year before the filing of the application that matured into the ’967 patent, also recognized that portions of pictures that will not be displayed do not need to be decoded. Specifically, *Chen* explained that “[t]hose skilled in the art will appreciate that the P-frames need only be partially decoded in order to recover the I-slices.” EX-1004 ¶[0025].

For the reasons discussed below, the Board should institute *inter partes* review of the ’967 patent and cancel claims 1-5 based on the disclosures of these references.

II. THE ’967 PATENT

A. Background

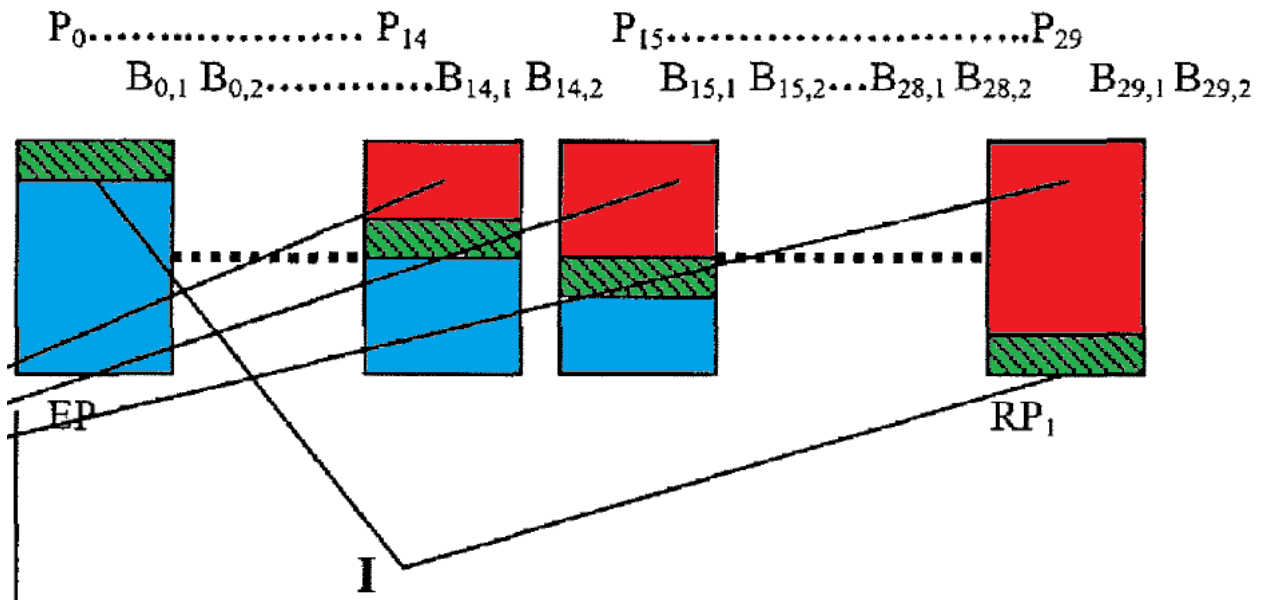
According to the background section of the ’967 patent, at the time the application that matured into the ’967 patent was filed, “distribution of digital video content for TV display [was] dominated by use of the MPEG-2 video compression standard (ISO/IEC 13818-2).” EX-1001 1:31-33. “An MPEG-encoded stream may have three types of pictures, Intracoded (I), Predicted (P) and Bi-directionally predicted (B).” *Id.* 1:35-37. Both the patent and *Chen* explain how these different types of pictures (or “frames”) differ from each other:

- 1) An intracoded picture (“I-picture”) is not compressed using any temporal predictions and can be decoded without the use of any other pictures,
- 2) A predicted picture (“P-picture”) uses temporal predictions from a picture that comes before it in the display order, so it can only be decoded with a picture that comes before it in the display order, either an I-picture or another P-picture, and
- 3) A bi-directionally predicted picture (“B-picture”) uses bi-directional motion predictions, so it can only be decoded with a picture that comes before it in the display order *and* a picture that comes after it in the display order.

EX-1001 1:35-47; EX-1004 ¶¶[0003], [0008].

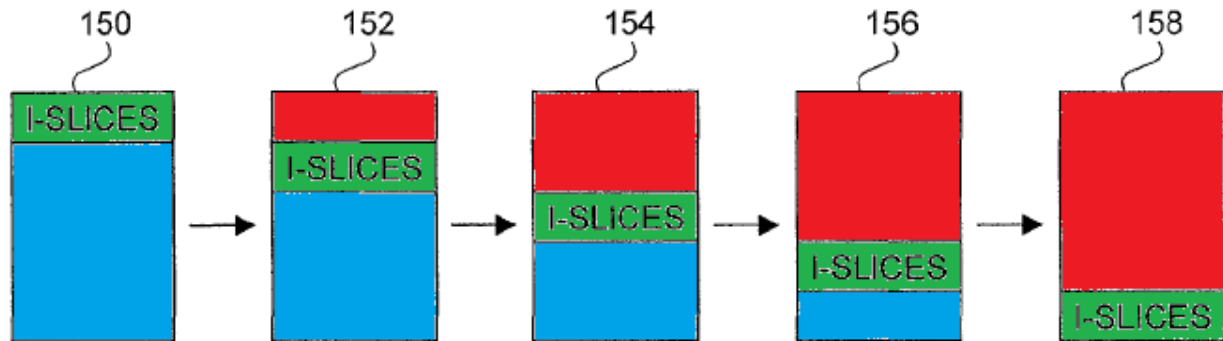
“During a Rewind operation, the pictures have to be displayed in the reverse order.” EX-1001 1:52-53. This “can be accomplished by decoding pictures in the forward picture order starting from the last I-picture before the B-picture in the forward decoding order. The I-picture is used as a reference picture since I-pictures do not require any other picture to be decoded.” *Id.* 1:61-65. As recognized by the background section of the ’967 patent, it was known that “the intervening pictures between the reference picture and the current picture only need to be decoded but not displayed.” *Id.* 1:65-67.

The background section of the '967 patent criticizes how existing decoders rewound “a special class of MPEG-2 streams, known as Headend In The Sky (HITS) streams.” EX-1001 2:1-37. Although the '967 patent thus admits that HITS streams were known, providing an extensive description of them in its background section, the patent repeats much of this description in its detailed description section. *Compare id. with id.* 5:37-64; *see Application of Nomiya*, 509 F.2d 566, 570 (C.C.P.A. 1975) *and Application of Hellsund*, 474 F.2d 1307, 1310-11 (C.C.P.A. 1973). Referring to Figure 4, reproduced below with colored annotations added, the patent explains HITS streams include “P-pictures, P, and B-pictures, B, but do not include I-pictures . . . because I-pictures require the most memory and bandwidth. Instead, HITS streams use a progressive refresh mechanism to build reference pictures,” with each P-picture having at least one intra-coded slice, I, where a slice comprises 16 horizontal lines of pixels. *Id.* 5:37-48; Fig. 4:

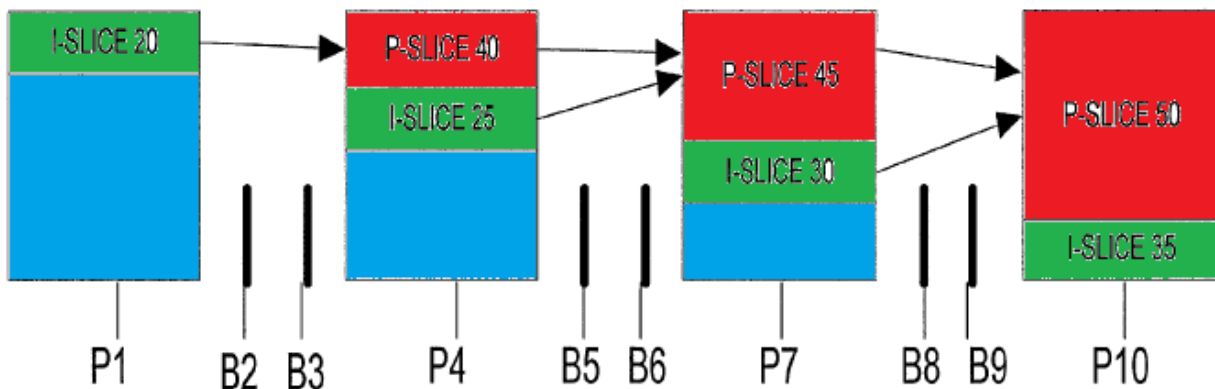


In Figure 4, each of a first P-picture P_{14} and a second P-picture P_{15} includes slices above the intra-coded slice (indicated in red), an intra-coded slice (shaded in original figure), and slices below the intra-coded slice (indicated in blue). *Id.* 5:37-45. The HITS stream is configured such that the intra-coded slice in a first picture, such as P_{15} , is vertically just below the intra-coded slice in a previous P-picture, such as P_{14} . *Id.* 5:48-50. This downward progression of intra-coded slices from picture to picture enables predicting slices above an intra-coded slice in a later picture (e.g., P_{15}) based on only the slices of the previous P-picture (e.g., P_{14}) that are above the current intra-coded slice. *Id.* 5:51-56. This “ensures that if a series of pictures is decoded starting from a P-picture whose first-slice is intra-coded, then a ‘clean’ refreshed picture will be built after all slices have been progressively refreshed.” *Id.* 5:56-59.

Both *MacInnis* and *Chen* describe progressive refresh mechanisms using figures similar to Figure 4 in the '967 patent. EX-1003 (*MacInnis*) ¶¶[0049]-[0058] and Fig. 4A (same highlighting as above):



EX-1004 (*Chen* '677) ¶¶[0022]-[0023], Fig. 1 (same highlighting):



The '967 patent summarizes the invention as a “system and method for enhancing performance of personal video recording (PVR) functions on HITS digital video streams.” EX-1001 2:60-65. According to the patent, “[d]uring rewind of a HITS stream, the video decoder builds a clean reference picture,” but “because the P-pictures are not displayed, the decoder does not decode the portion

of the P-picture below the last intracoded slice.” *Id.* 2:65-3:3. This “advantageously reduces the processing required to build a clean reference picture,” and is possible “because the subsequent pictures do not use the said portions for prediction.” *Id.* 3:3-10.

B. Challenged Claims

LGE challenges claim 1 and its dependent claims 2-5.

1. Claim 1

1. [1.0] A system for displaying pictures, said system comprising:

[1.1] a host processor for transmitting transport packets, said transport packets providing a plurality of instructions;

[1.2] a video decoder for executing the plurality of instructions; and

[1.3] wherein execution of the instructions by the video decoder comprises causing:

[1.4] selecting a picture, said picture comprising

[1.4.a] an intracoded slice and at least one slice above the intracoded slice, and

[1.4.b] at least one slice below the intracoded slice;

[1.5] decoding the at least one slice above the intracoded slice;

[1.6] decoding the intracoded slice; and

[1.7] decoding at least a portion of another picture after decoding the at least one slice above the intracode slice and the intracode slice without having decoding the at least one slice below the intracoded slice.

2. Claim 2

2. [2.0] The system of claim 1,

[2.1] wherein the another picture comprises an intracoded slice and at least one slice above the intracoded slice, and

[2.2] wherein decoding at least a portion of the another picture further comprises:

[2.2a] decoding the at least one slice above the intracoded slice in the second picture; and

[2.2b] decoding the intracoded slice of the second picture.

3. Claim 3

3. The system of claim 2, wherein causing decoding the at least one slice above the intracoded slice in the another picture comprises causing: predicting the slices above the intracoded slice in the another picture from the intracoded slice and the slices above the intracoded slice in the picture.

4. Claim 4

4. The system of claim 1, wherein decoding the at least one slice above the intracoded slice further comprises causing: decoding the at least one slice in raster order.

5. Claim 5

5. The system of claim 1, wherein the picture comprises a P-picture from a HITS stream.

C. Relation to CIP Parent Application

The application that matured into the '967 patent is a continuation-in-part (CIP) of U.S. Application No. 09/651,693, filed on September 12, 2001. EX-1006 (copy of originally filed application from USPTO File Wrapper). The CIP parent application does not disclose all of the features of the claims in the '967 patent, so these claims are not entitled to the filing date of the CIP parent application.

PowerOasis, Inc. v. T-Mobile USA, Inc., 522 F.3d 1299, 1306-07 (Fed. Cir. 2008).

For example, the CIP parent application lacks any discussion of “decoding at least a portion of another picture after decoding the at least one slice above the intracode slice and the intracode slice *without having decoding the at least one slice below the intracoded slice*,” as required by claim 1. EX-1002 ¶¶35-39. The slices in the CIP parent application are used to determine whether a frame is an Entry Point Picture (EP), a portion of a stream “that would efficiently allow a complete picture to be decoded.” EX-1006 25:14-22; *see also* 26:15-28:8. There is

no discussion of decoding only particular slices in one picture and then decoding an additional portion of another picture.

To the extent that Broadcom asserts, before or after institution, that the CIP parent application's appendix supports the claims in the '967 patent, LGE notes that it does not. EX-1002 ¶¶35-39. The appendix describes a progressive refresh mode in which "I-pictures are not mandated in the bit stream" and points out a limitation of this mode is that "motion vectors for the macro-blocks located above the refreshed slices in the current P-picture can only point to the region above refreshed slices in the previous P-picture." *See* EX-1006 46:2-19.

But this limitation has nothing to do with whether any of the slices in the previous P-picture are decoded. To the contrary, a POSITA would appreciate that it merely prevents macro-blocks located above refreshed slices in a P-picture from being decoded based on certain P-slices in a previous P-picture. EX-1002 ¶¶37-38. Thus, the CIP parent application's appendix does not support claim 1's "decoding at least a portion of another picture . . . *without having decoding* the at least one slice below the intracoded slice."

Accordingly, neither claim 1 nor its dependent claims 2-5 are entitled to the filing date of the CIP parent application. To the contrary, they are only entitled to the filed date of the application that matured into the '967 patent: December 11, 2002.

III. LEVEL OF ORDINARY SKILL

A person of ordinary skill in the art (“POSITA”) as of December 11, 2002 (the earliest possible effective filing date of the ’967 patent) would have had at least a Bachelor’s degree in Electrical Engineering, Computer Engineering, or Computer Science, and three to four years of experience in image/video processing or graphics systems, or an equivalent degree and/or experience, where more education could substitute for experience. EX-1002 ¶41.

IV. CLAIM CONSTRUCTION¹

A claim in an unexpired patent subject to *inter partes* review receives the “broadest reasonable construction in light of the specification of the patent in which it appears.” 37 C.F.R. § 42.100(b). The broadest reasonable construction should be applied to all claim terms in the ’967 patent. LGE submits that, for the purposes of this petition, no explicit construction is needed for any claim term not addressed below.

¹ LGE has not included any indefiniteness arguments because the IPR procedure does not permit them, but LGE may raise such arguments in other proceedings.

A. “instructions” (claim 1)

For the purposes of this petition, the term “instructions” means commands, separate from video or audio data, that specify trick play functions to be performed. EX-1002 ¶¶47-50.

The '967 patent differentiates the parts of a transport stream (TS) including video or audio data from the parts of the TS including instructions. According to the patent, a data transport processor 335 “passes the audio TS to an audio decoder 360 and the video TS to a video transport processor 340, and then to a MPEG video decoder 345 that is operable to decode and extract embedded, TS formatted command packets, that may include instructions to perform trick play functionality.” EX-1001 5:6-13. In addition to these TS formatted command packets, the “video TS comprises pictures that are compressed representations of individual images forming a video.” *Id.* 5:25-26.

The CIP parent application of the '967 patent, which is incorporated by reference into the '967 patent, EX-1001 1:8-12, 7:37-43, also makes clear that the claimed “instructions” are separate from any audio or video data. The CIP parent explains that

The MPEG TS content 400 includes an MPEG TS 410.

The MPEG TS 410 includes *data 420* and *TS formatted trick play commands 430*. The data 410 may include any

one or combination of *video data 422, audio data 424, . . . , and any other data type 429* as understood by those persons having skill in the art. The *TS formatted trick play commands 430* may include any one or *combination of commands* that direct functions of pause/still 431, fast forward 432, rewind 433, slow forward 434, slow rewind 436, skip 436, . . . , and any other *commands 439* as understood by those persons having skill in the art.²

EX-1006 ¶[0051]. In a related litigation, ITC-337-TA-1047, Broadcom asserts that “instructions” means “the contents of the coded or compressed audio or video syntax itself, which instructs or causes the decoder to perform various tasks.”

EX-1008 3. But the patent makes clear that the broadest reasonable construction *does not* require that the “instructions” be contents. Rather, as explained in both the ’967 patent and its CIP parent application, “instructions” are separate and different from “data” that makes up the audio/video stream. EX-1001 5:9-14, 25-28;

EX-1006 ¶[0051].

² All emphasis is added unless otherwise indicated.

B. “without having decoding the at least one slice below the intracoded slice” (claim 1)

For the purposes of this petition, the term “without having decoding the at least one slice below the intracoded slice” means omitting decoding the at least one slice below the intracoded slice. EX-1002 ¶¶51-53.

The ’967 patent explains that “[d]uring rewind of a HITS stream, the video decoder builds a clean reference picture.” EX-1001 2:65-67. The reference pictures are built by decoding each P-picture between two Entry Pictures (EPs). *Id.* 1:67-2:1. However, the ’967 patent explains that the P-pictures are not displayed, so “the decoder does not decode the portion of the P-picture below the last intracoded slice.” *Id.* 3:1-3. The patent explains that “[o]mitting decoding the portion of the P-pictures below the intracoded slice advantageously reduces the processing required to build a clean reference picture.” EX-1001 3:8-10.

C. “transport packet” (claim 1)

For the purposes of this petition, the term “transport packet” means a packet containing either data or instructions.

According to the CIP parent application of the ’967 patent, which is incorporated by reference into the ’967 patent, EX-1001 1:8-12, 7:37-43, a “transport stream” can contain “both data packets and command packets.”

EX-1006 ¶[0011]. “Transport packet” in this context thus refers to a portion of a stream that contains one of data or commands. EX-1002 ¶¶54-55.

V. CLAIMS 1-5 OF THE '967 PATENT ARE UNPATENTABLE OVER THE PRIOR ART

Claims 1-5 are anticipated under 35 U.S.C. § 102 and/or obvious under § 103. *MacInnis* anticipates claims 1-5. *Infra* at Section V.A. *MacInnis* in view of *ISO 13818-2* renders obvious claim 4. *Infra* at Section V.B. *Chen '677* anticipates claims 1-4. *Infra* at Section V.C. *Chen '677* renders obvious claim 5. *Infra* at Section V.D. And *Chen '677* in view of *ISO 13818-2* renders obvious claims 1-5. *Infra* at Section V.E.

A. Ground 1: *MacInnis* anticipates claims 1-5.

U.S. Patent Application Publication No. 2002/0061183 to *MacInnis* (“*MacInnis*”) was filed on October 9, 2001, and published on May 23, 2002. EX-1003. *MacInnis* also claims priority to provisional application no. 60/239,425, filed October 10, 2000. EX-1005. *MacInnis* is prior art under at least 35 U.S.C. § 102(a). *Supra* at Section II.D.

To the extent that Broadcom asserts or the Board finds, before or after institution, that the challenged claims of the '967 patent are entitled to the filing date of the CIP parent application (which they are not), *MacInnis* would still be

prior art under at least § 102(e) because the provisional application provides written description support for at least claim 1 of *MacInnis*:

<i>MacInnis</i> , claim 1	Provisional Application No. 60/239,425
1. A method for enabling trick modes for a digitally encoded stream, the method comprising the steps of:	26:4-5 (“1. A <i>method for enabling trick modes</i> in a personal video recording system, <i>the method comprising the steps of:</i> ”); 2:19-21 (“present invention provides a system and method for PVR to record <i>digitally encoded data streams</i> , and to <i>playback and decode the recorded streams</i> ”)
storing the digitally encoded stream, wherein the digitally encoded stream comprises video pictures;	19:30-31 (“In another embodiment, transport streams (TS) <i>are stored on disk and played from disk.</i> ”); 26:6-7 (claim 1) (“receiving a <i>digitally encoded stream including video pictures</i> for display at a display rate”)
decoding the video pictures, starting from an entry picture, to generate one or	26:8-9 (claim 1) (“ <i>decoding the video pictures</i> at a rate faster than the display

more fully decoded pictures, wherein the entry picture is a non-intra-coded picture; and	rate”), 3:6-9 (“Reverse decode and display may be implemented by <i>starting at a suitable entry point, decoding quickly up to the point of the desired picture, and displaying the result, and repeating the process.</i> ”); 2:24-27 (“The invention . . . may be used with streams that <i>have no I pictures and that have a ‘progressive refresh’ pattern of using intra-slices (I-slices)</i> ”); 5:30-34 (“if the decoding starts with a picture whose first row is an I-slice, then successive P pictures can be decoded until a picture is decoded where the bottom row is an I-slice, and from that picture on, the <i>resulting pictures are fully decoded and capable of being presented.</i> ”)
displaying at least one of the fully decoded pictures.	26:11-12 (claim 1) (“ <i>display video pictures</i> that are aligned with available

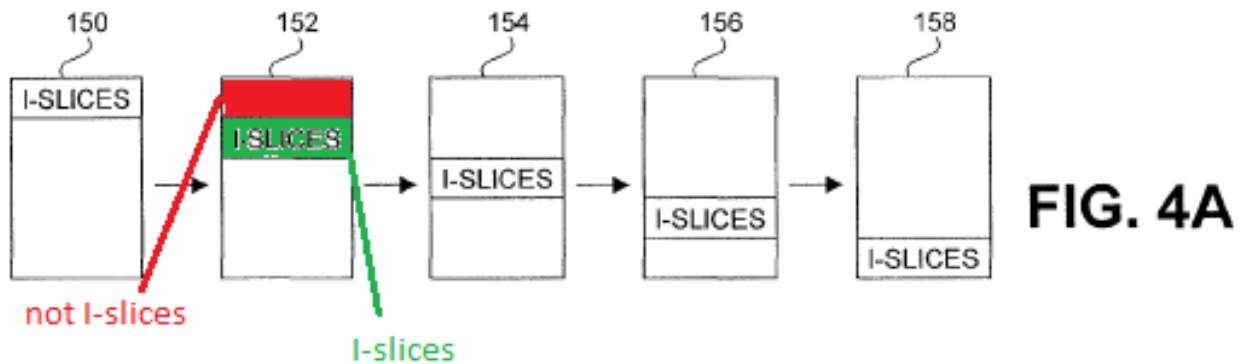
	display times”) ; 5:30-34 (“if the decoding starts with a picture whose first row is an I-slice, then successive P pictures can be decoded until a picture is decoded where the bottom row is an I-slice, and from that picture on, the <i>resulting pictures are fully decoded</i> and capable of being presented.”)
--	---

EX-1002 ¶58.

MacInnis discloses a system and method for recording and playback of digitally encoded audio/video streams. EX-1003 ¶[0002]. Like the ’967 patent, *MacInnis* relates to decoding digital streams without I-pictures, such as progressive refresh or HITS (“Headend In The Sky”) streams. EX-1003 ¶[0008], [0050]; *see also* EX-1001 2:1-6. These known types of streams may also be referred to as progressive refresh I-slice-based streams. EX-1003 ¶[0046]. These streams include P pictures and B pictures, but no I pictures. *Id.* ¶[0052]-[0053].

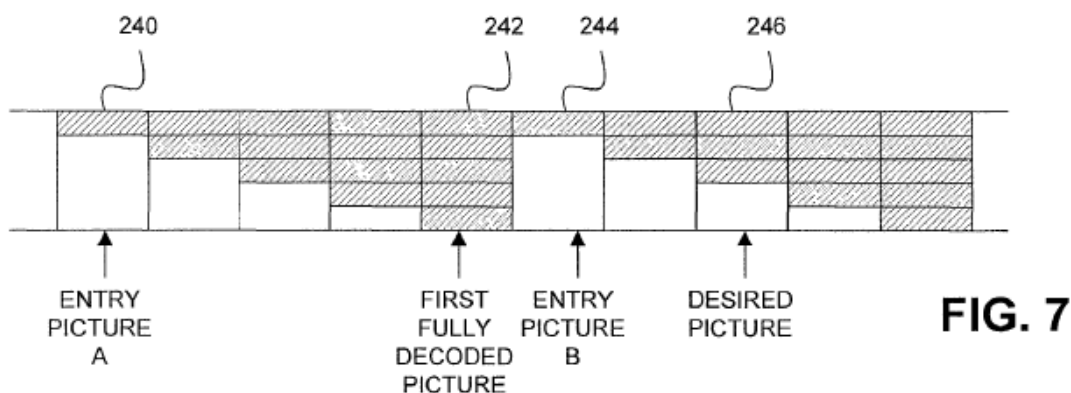
For example, Fig. 4A (annotated below) depicts pictures 150, 152, 154, 157, and 158. A “refresh depth” that defines the number of rows of I-slices in each P-picture may be used to define how many rows of each picture are I-slices. For

example, the “first D rows” (D being an integer representing refresh depth) of picture 150 are intra-coded, i.e., I-slices. EX-1003 ¶[0056]. The “remaining rows typically are predictively coded, i.e., P-slices.” *Id.* In picture 152, “the first D rows are *not* I-slices and the second D rows are [I-slices].” *Id.* ¶[0057].



The decodable (or refreshable) portion of each picture, the I-slices, moves down “until all rows in the consecutive sequence of pictures [] [have] been represented in I-slices.” *See id.* ¶¶[0056]-[0057] (discussing pictures 150, 152, 154, 156, 158), Fig. 3 (note that I-slices move down from picture to picture).

Figure 7 depicts a “timing diagram” for decoding pictures having I-slices in progressive bitstreams. ¶[0021]. This figure illustrates the “reverse play” or “rewind” trick play process. ¶¶[0108]-[0109].



During a rewind process, where the pictures in Fig. 7 must be displayed quickly from right to left (i.e., in reverse), pictures must be decoded from left to right, from EP (Entry Picture) to EP. *See, e.g., id.* ¶[0107]:

reverse display involves *starting at entry pictures, decoding forward until the desired fully decoded picture is reached, and displaying the resulting picture* while again decoding a portion of the stream, stopping the decode process at an earlier picture, and possibly starting the decode process from an earlier coded picture if necessary to achieve a fully decoded picture.

For example, to obtain a desired picture 246, the decoding process may begin by decoding Entry Picture A 240. ¶[0110]. After decoding EP A 240, the decoder continues decoding the pictures to the right of EP A 240 (including pictures 242 and 244), until reaching and decoding desired picture 246. *Id.* ¶[0111]. This decoding process starts with EP A 240, and not EP B 244, because the lowest two slices in desired picture 246 may be non-intracoded and thus cannot be decoded if

the decoding process starts at EP 244. *See id.* ¶¶[0021], [0056], [0111];

EX-1002 ¶64. Thus, starting decoding at EP A 240 enables decoding of slices below the last I-slice in desired picture 246. *Id.* ¶[0110].

As explained by *MacInnis*, because the I-slices are encoded in a progressive pattern, *see* ¶¶[0021], [0056], “not all slices [in every picture] must be decoded.” *Id.* ¶[0111]. Instead, *only* “necessary slices” are decoded, which can double the average decoding rate. *Id.*

Fig. 3 depicts playback system 100 which enables playback of decoded MPEG video data:

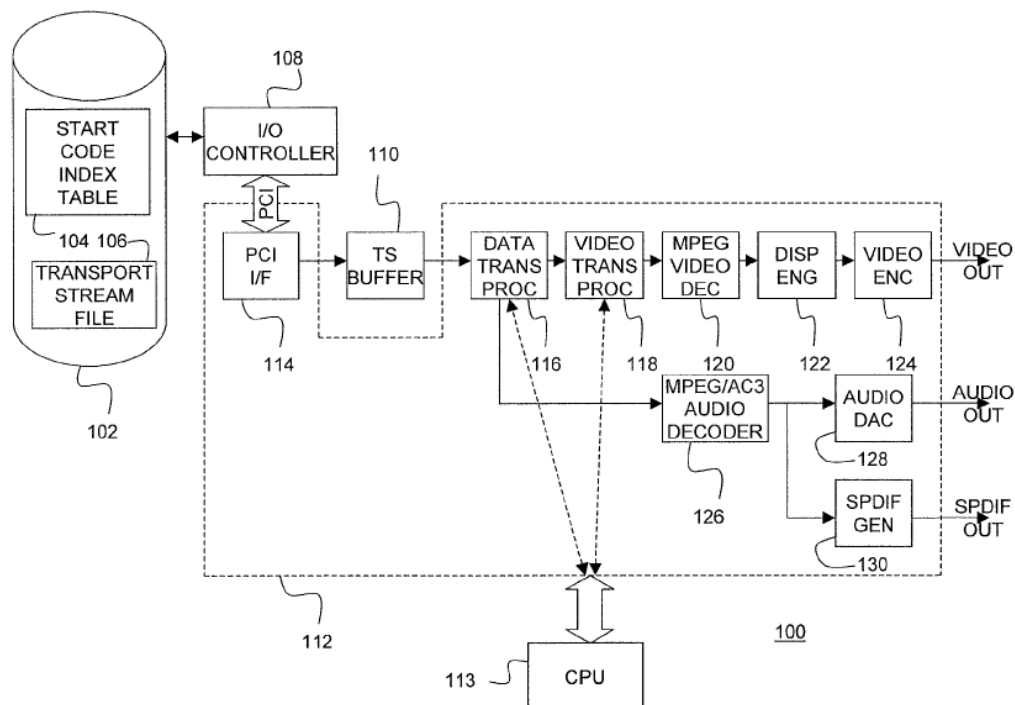


FIG. 3

CPU 113 reads a transport stream (“TS”) from storage device 102 into a TS buffer 110. *Id.* ¶[0036]. Similar to the host processor in the ’967 patent, CPU 113 may control a data transport processor 116 and/or a video transport processor 118 to implement trick play such as rewinding. *Id.* ¶[0037]. *Compare* EX-1003

¶[0037]:

CPU 113 may control the data transport processor 116 and/or the video transport processor 118 to implement trick modes (e.g., pause, fast forward, slow forward, rewind, fast reverse, slow reverse, skip).

with EX-1001 3:45-50:

The TS 115 may be received by the decoder 120 from a host processor 110, or any other source 105 without departing from the scope and spirit of the invention. The *host processor 110* or the any other source 105 is the *device controlling the playback* (including *trick play* playback) of the data.

Data transport processor 116 in *MacInnis* also operates similarly to data transport processor 335 described in the ’967 patent, by demultiplexing a transport stream TS and passing the constituent parts to respective decoders. *Compare* EX-1003 ¶[0037] (“data transport processor 116 preferably de-multiplexes the TS into its constituents and *passes...MPEG video (video Transport Stream) to a video*

transport processor 118 and then to an MPEG video decoder 120”) with EX-1001 5:7-15 (“data transport processor [335] then de-multiplexes the TS into its PES constituents and passes...the video TS to a video transport processor 340, and then to a MPEG video decoder 345 that is operable to decode and extract embedded, TS formatted command packets”). EX-1002 ¶68.

As further detailed below, *MacInnis* anticipates claims 1-5 under the broadest reasonable constructions consistent with the ’967 patent or those offered by Broadcom in the related ITC Investigation. EX-1008.

1. Claim 1

a. Element [1.0]: *MacInnis* discloses “[a] system for displaying pictures.”

To the extent that Broadcom argues or the Board finds, before or after institution, that the preamble limits the claims, *MacInnis* discloses it. *MacInnis* discloses a system and method for recording and playback of digitally encoded audio/video streams. EX-1003 ¶[0002]. Fig. 3 depicts playback system 100:

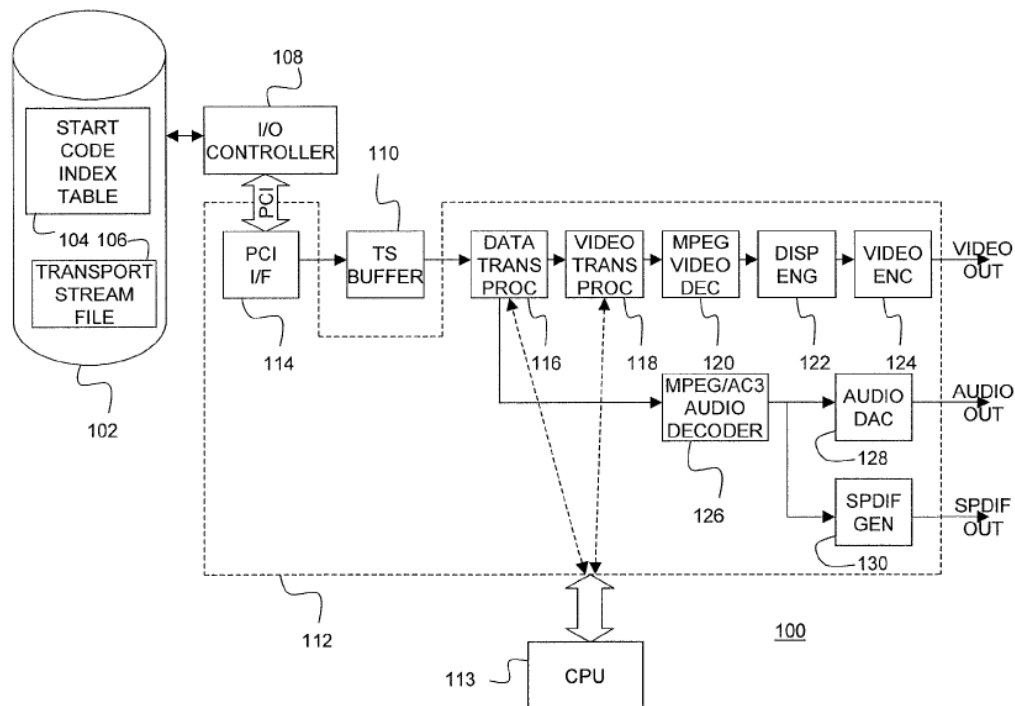


FIG. 3

Playback system 100 includes CPU 113. CPU 113 reads a transport stream (TS) from a storage device 102 and a data transport driver in CPU 113 is “responsible for data flow of the stream to be played back through the data transport processor 116 and/or the video transport processor 118.” *Id.* ¶[0040]. CPU 113 “may control the data transport processor 116 and/or the video transport processor 118 to implement trick modes” such as “rewind.” *Id.* ¶[0037].

The “data transport processor 116 de-multiplexes the TS into its constituents.” *Id.* ¶¶[0036]-[0037]. Data transport processor 116 transmits the MPEG video TS to a video transport processor 118 and an MPEG video decoder 120. ¶[0037]. The MPEG video decoder may decode the MPEG video and send it

to a display engine 122. ¶[0045]. Display engine 122 may scale the video, render graphics, and construct a complete display based on the decoded video. *Id.* Once the video is “ready to be presented,” display engine 122 may pass the video to a video encoder 124 for output in analog format or in digital format. *Id.*

Thus, because playback system 100 may read and pass a stream to a video encoder for output, *MacInnis*’ playback system 100 discloses the claimed “system for displaying pictures.”

b. Element [1.1]: *MacInnis* discloses “a host processor for transmitting transport packets, said transport packets providing a plurality of instructions.”

As discussed above, *supra at* Section IV.C, the term “transport packet” means a packet containing either data or instructions. Moreover, the term “instructions” means commands, separate from video or audio data, that specify trick play functions to be performed, *id.*, while Broadcom’s construction of this term is “the contents of the coded or compressed audio or video syntax itself, which instructs or causes the decoder to perform various tasks.” *Id.*, *see also* EX-1008 3. Because the meaning of “transport packet” includes the term “instructions,” the term “transport packet” means one of two things:

- I. a packet containing either data or commands, separate from video or audio data, that specify trick play functions to be performed; and

- II. a packet containing either data or the contents of the coded or compressed audio or video syntax itself, which instructs or causes the decoder to perform various tasks.

Under either, *MacInnis* teaches element [1.1].

i. Construction I

CPU 113 in *MacInnis* discloses the claimed host processor for transmitting transport packets providing a plurality of instructions under the first construction of “transport packet.” Broadcom has asserted in related litigation, ITC Investigation No. 337-TA-1047, that the phrase “host processor” means

any device or element that processes an incoming transport stream (i.e., a stream of packets), extracts the packets from the stream, and passes the packets on to a decoder device (such as any audio or visual decoding device or functionality),

citing 3:45-52 of the '967 patent. EX-1008. Under at least this construction, CPU 113 in *MacInnis* teaches the claimed “host processor.” CPU 113 “reads the TS [(transport stream)] data from storage device 102 based on the user selected playback mode.” ¶[0036]. Reading this transport stream comprises extracting the packets from it. *See* ¶[0027] (explaining that the TS comprises “packets”). CPU 113 may comprise a data transport driver, or “DTD.” ¶[0040]. DTD may be

responsible for playing back the stream “through the data transport processor 116 and/or the video transport processor 118.” *Id.* ¶[0037].

The DTD may control the playback of the stream by “*insert[ing] a command transport packet* containing various commands to the video transport processor 118 and video decoder 120.” *Id.* ¶[0040]. Inserting command transport packets comprises inserting them into the transport stream and transmitting them to decoder 120. *Id.*; EX-1002 ¶88.

Because the DTD inside of CPU 113 inserts into the stream transport packets that contain commands, separate from video or audio data, that specify trick play functions to be performed (i.e., “instructions”) into a stream for transmission to video transport processor 118 and video decoder 120, CPU 113 teaches element [1.1] under the first construction. EX-1002 ¶89.

ii. Construction II

Should Broadcom argue or the Board find, before or after institution, that “instructions” is defined by the second construction (i.e., “the contents of the coded or compressed audio or video syntax itself, which instructs or causes the decoder to perform various tasks”), CPU 113 in *MacInnis* meets this limitation as well.

DTD in CPU 113 is “responsible for data flow of the stream to be played back through the data transport processor 116 and/or the video transport processor 118.” EX-1003 ¶[0040]. CPU 113 “[feeds]” the video stream to the decoder. *Id.*

¶[0099]. The video stream itself constitutes the “contents of the coded or compressed audio or video syntax itself, which instructs or causes the decoder to perform various tasks” under Broadcom’s construction. EX-1002 ¶91.

Because CPU 113 transmits the video stream to the decoder, *MacInnis* teaches element [1.1] under this construction as well.

c. Element [1.2]: *MacInnis* discloses “a video decoder for executing the plurality of instructions.”

MacInnis discloses element [1.2]. *MacInnis* discloses, as discussed above, an MPEG video decoder 120. EX-1003 Fig. 3, ¶[0037]. The DTD in CPU 113 may control the playback of the stream by “insert[ing] a command transport packet *containing various commands to . . . video decoder 120.*” ¶[0040].

Video decoder 120 may use these commands to control the data flow of the MPEG video stream, for example, by implementing and controlling trick modes. ¶[0043]. Trick modes include, for example, pausing the stream, fast forwarding the stream, rewinding the stream, or skipping portions of the stream. ¶[0037].

Video decoder 120 supports these trick modes during playback by “randomly access[ing] the recorded Transport Stream (e.g., TS file 106) stored in storage device 102 to retrieve the correct TS data.” ¶[0038]. In particular, video decoder 120 interprets the command fields in the transport packet inserted into the stream to control the MPEG video stream. ¶¶[0040], [0043]; EX-1002 ¶95.

Because video decoder 120 receives command transport packets that contain various commands, and uses those commands to operate on the MPEG stream, for example, to effect trick play, *MacInnis*' video decoder 120 constitutes the claimed "video decoder for executing the plurality of instructions." EX-1002 ¶96.

d. Element [1.3]: *MacInnis* discloses "wherein execution of the instructions by the video decoder comprises causing"

As discussed above with respect to element [1.2], video decoder 120 interprets command fields in transport packets inserted into the MPEG stream to control the MPEG video stream. EX-1003 ¶[0040]. In particular, video decoder 120 can execute these commands to perform trick modes during playback. EX-1003 ¶¶[0038], [0043]; EX-1002 ¶97.

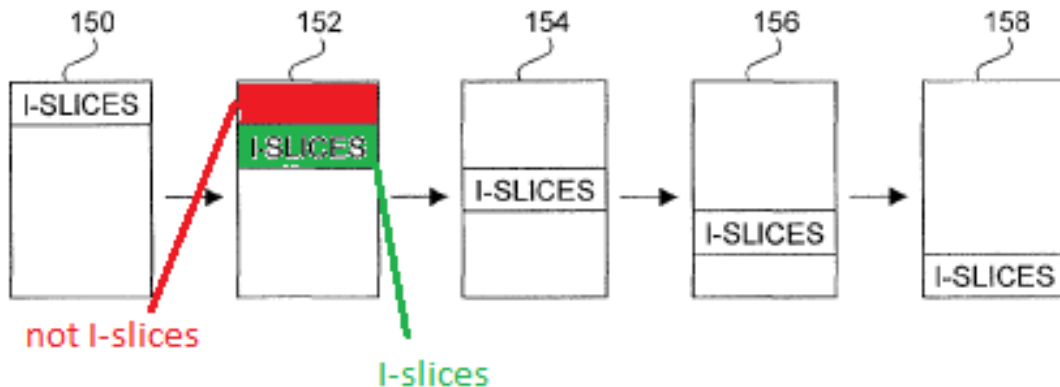
Because video decoder 120 receives command transport packets that contain various commands, and executes those commands to operate on the MPEG stream, *MacInnis*' video decoder 120 teaches executing "the instructions by the video decoder."

e. Element [1.4]: *MacInnis* discloses "selecting a picture, said picture comprising an intracoded slice and at least one slice above the intracoded slice, and at least one slice below the intracoded slice."

MacInnis discloses element [1.4], which is broken into subparts below due to its length.

- i. **Element [1.4.a]: *MacInnis* discloses “selecting a picture, said picture comprising an intracoded slice and at least one slice above the intracoded slice.”**

Figure 4A of *MacInnis* depicts pictures 150-158 in a stream that does not have I-pictures. EX-1003 ¶¶[0047]-[0050], [0056]. For example, in picture 152, the “first D [(D being an integer)] rows are not I-slices, and the second D rows are [I-slices].” ¶[0057].



Picture 152 constitutes the claimed “picture.” EX-1002 ¶101. *MacInnis* explains that decoding a stream for trick play functionality comprises “starting at entry pictures” and “decoding forward until the desired fully decoded picture is reached.” EX-1003 ¶[0107]. Thus, *MacInnis* teaches selecting each picture as part of a decoding process. EX-1002 ¶101.

MacInnis explains that within P-pictures (such as P-pictures 150-158) “there are some rows that are coded as intra-slices (I-slices).” ¶[0054]. The remaining

rows of each picture “typically are predictively coded, i.e., P-slices, even though it is possible for the remaining rows to include I-slices.” *See* EX-1003 ¶[0056].

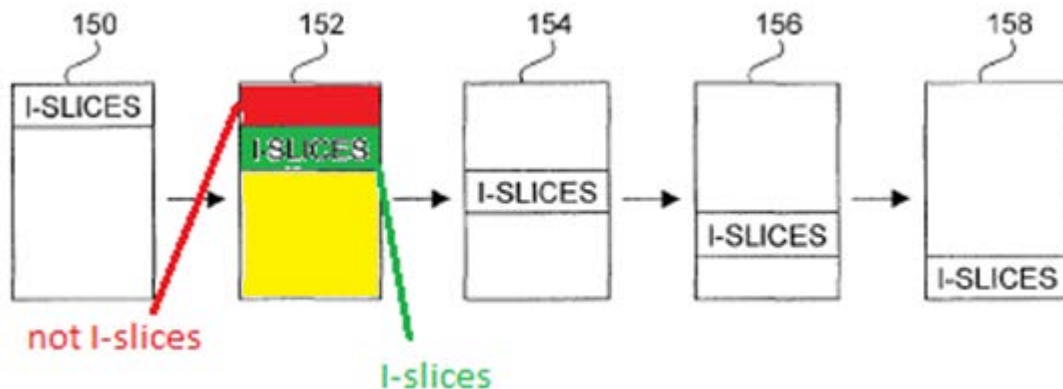
MacInnis makes clear that “D” may be an integer that represents the number of rows that “are encoded as I-slices in each P-picture” (the “refresh depth”). ¶[0055]. The “I-slices” portion may contain “[a]nywhere from 1 to 9 rows [] encoded as I-slices.” *Id.* Thus, the “I-slices” may be a single row encoded as an I-slice. *Id.*

The “I-slices” constitutes the claimed “intracoded slice,” and the rows above the intracoded slice constitute the claimed “at least one slice above the intracoded slice.” EX-1002 ¶104.

ii. Element [1.4.b]: *MacInnis* discloses “said picture comprising...at least one slice below the intracoded slice.”

As discussed above with respect to element [1.4.a], picture 152 contains rows that are not coded as intracoded slices (in red) and rows that are coded as intracoded slices (in green). EX-1003 ¶¶[0054], [0057].

MacInnis explains that the “remaining rows” that follow rows coded as intracoded slices “typically are predictively coded, i.e., P-slices.” ¶[0056]. Each picture in the sequence of pictures includes predictively coded slices below the I-slices; these “remaining rows” in picture 152 are highlighted below in yellow:

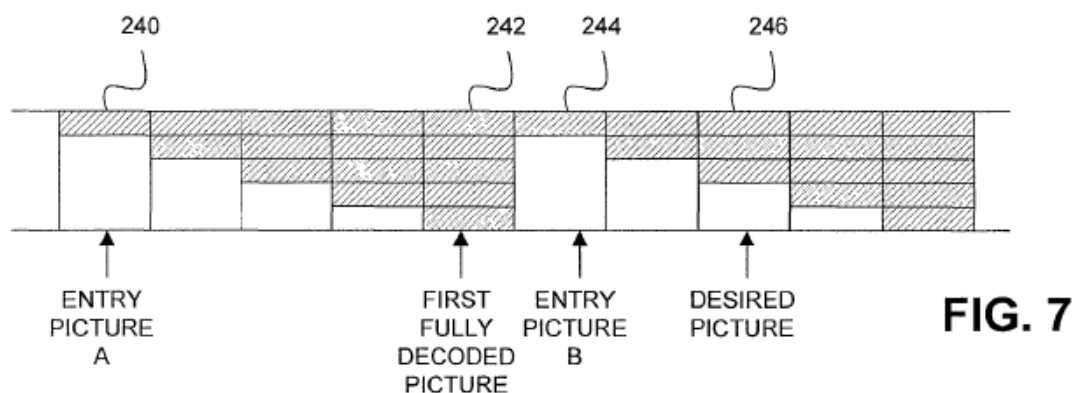


The rows that follow the I-slices in *MacInnis* constitute the claimed “at least one slice below the intracoded slice.” EX-1002 ¶107.

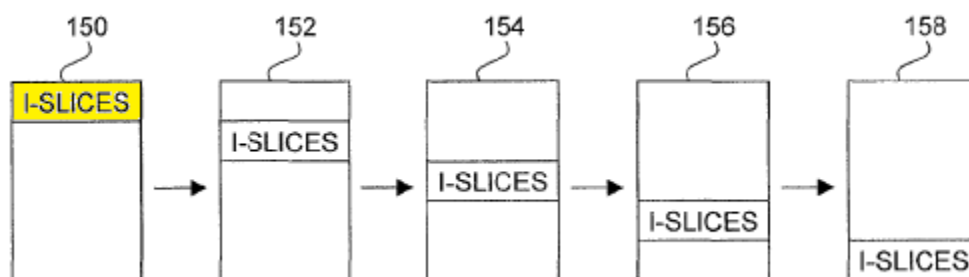
f. Element [1.5]: *MacInnis* discloses “decoding the at least one slice above the intracoded slice.”

MacInnis discloses element [1.5]. *MacInnis* explains that decoding starts with a picture whose first row is an I-slice, and may continue successively “until a picture is decoded where the bottom row is an I-slice.” ¶¶[0060], [0058]. For example, decoding may start at picture 150 (where “the first D rows...are I-slices”) and continue until picture 158 (“where the bottom row is an I-slice”). *See, e.g., id.* ¶¶[0056]-[0060].

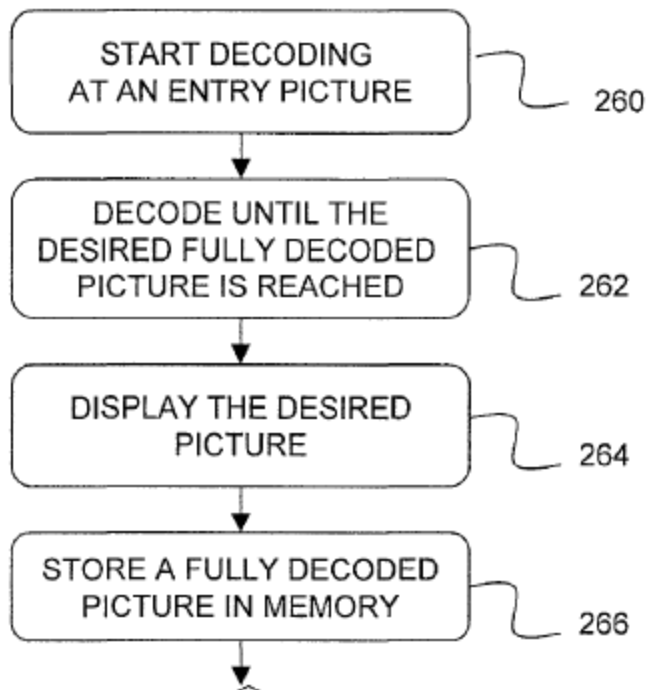
Fig. 7 illustrates a process of implementing “reverse display” such as trick play rewind. EX-1003 ¶[0107]:



Decoding a stream for trick play rewind, where pictures are displayed from right to left, starts by decoding pictures in forward order, e.g., from left to right, starting at EP 240 (Entry Picture). *See, e.g., id.* ¶[0107]: “reverse display involves starting at *entry pictures*, *decoding forward until the desired fully decoded picture is reached*, and displaying the resulting picture.” Entry Pictures (EP) are pictures whose first row is an I-slice. *Id.* ¶[0061]. Because picture 150 in Fig. 4A has I-slices for its topmost D rows, picture 150 is an EP:

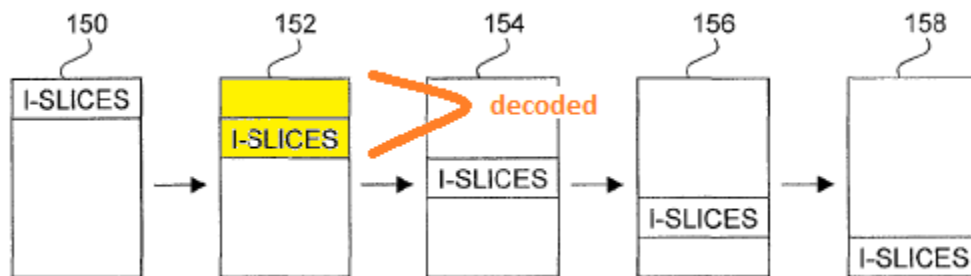


After decoding the EP, “the process proceeds with decoding forwards from the entry picture until the desired picture has been decoded in step 262, and [] the result [is displayed] in step 264.” *Id.* ¶[0111]; see also Fig. 8:



This means that pictures are decoded from left to right (Fig. 7: from EP A 240 to desired picture 246). *See, e.g., id.* ¶[0107], Fig. 7.

Thus, after decoding picture 150 where the topmost D rows are I-slices, ¶[0056], MPEG video decoder 120 in playback system 100 begins decoding picture 152, where the topmost D rows are not I-slices, but the second set of D rows are I-slices. ¶[0057], Fig. 4A:

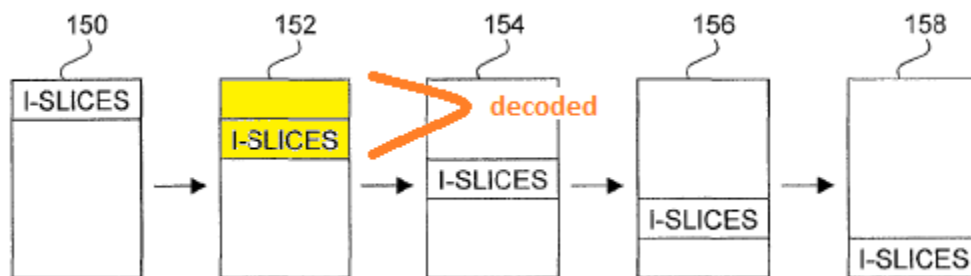


Indeed, ¶[0095] explains that “for the entry picture [(picture 150)], only the I-slices (rows) at the top of the picture preferably are decoded” and for the “second picture [(picture 152)] the (2xrefresh depth) [(2*D)] rows at the top [of the picture] preferably are decoded.”

Because the first D rows of the claimed “picture” (e.g., picture 152) are decoded, and the first D rows include slices above the I-slices, *MacInnis* discloses the claimed “decoding the at least one slice above the intracoded slice.”

g. Element [1.6]: *MacInnis* discloses “decoding the intracoded slice.”

MacInnis discloses element [1.6]. As discussed above with respect to element [1.5], MPEG video decoder 120 may decode picture 152, where the topmost D rows are not I-slices, but the second set of D rows are I-slices. ¶[0057]. ¶[0095] explains that for this second picture, the first D *and* second D rows “at the top [of the picture] preferably are decoded.” *See again* Fig. 4A:



Because the second D rows of the claimed “picture” 152 are decoded, and the second D rows include I-slices, *MacInnis* discloses the claimed “decoding the intracoded slice.”

- h. Element [1.7]: *MacInnis* discloses “decoding at least a portion of another picture after decoding the at least one slice above the intracode slice and the intracode slice without having decoding the at least one slice below the intracoded slice.”**

MacInnis discloses element [1.7]. As discussed above with respect to elements [1.4]-[1.6], *MacInnis* explains that decoding starts with a picture whose first row is an I-slice, and may continue successively “until a picture is decoded where the bottom row is an I-slice.” ¶[0060]. For example, decoding may start at picture 150 (where “the first D rows...are I-slices”), continue to picture 152 (for which “the first D rows are not I-slices, and the second D rows are”), and then continue to picture 154 (for which the “first 2*D rows are not I-slices, and the 3rd D rows are”). *See, e.g., id.* ¶¶[0056]-[0060]. Thus, picture 154 constitutes the claimed “another picture.” Moreover, for the reasons discussed above with respect to elements [1.5] and [1.6], *MacInnis* teaches “decoding the at least one slice above the intracode slice and the intracode slice” as also recited in element [1.7].

MPEG video decoder 120 may decode picture 154. *See, e.g.,* ¶[0107] (discussing left-to-right decoding of stream). Thus, *MacInnis* discloses “decoding

at least a portion of another picture after decoding the at least one slice above the intracode slice and the intracode slice.” EX-1002 ¶120.

MacInnis explains that not all slices of each picture must be decoded. To save on processing time, *MacInnis* explains that during fast-forward trick play, only the I-slices and the rows above the I-slices must be decoded, and that other slices may be omitted from the decoding process:

in a progressively refreshed bitstream where the location of the I-slices moves from top to bottom, for the entry picture, *only the I-slices (rows) at the top* of the picture preferably are decoded. For the second picture, *the (2 x refresh depth) rows at the top* preferably are decoded, and so on until I-slices have been decoded at the bottom of the last picture of the BOP.

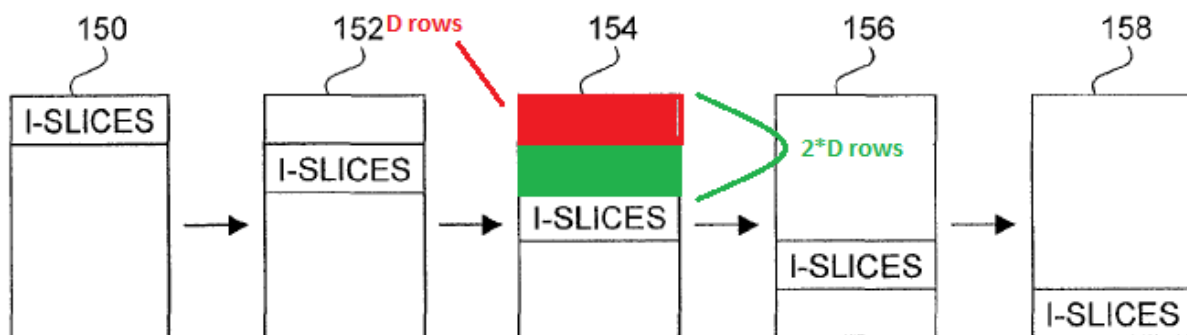
EX-1003 ¶[0095]. This means that each picture is decoded from the top of the picture until the end of the I-slice. The decoder thus may omit the slices that come below the I-slices and “only” decode the I-slices (rows) at the top of the picture.”

EX-1003 ¶[0095]; EX-1002 ¶122. This process of decoding only particular slices applies equally to the reverse process:

not all slices need to be decoded, *just as in the case of fast forward* at high rates while skipping data. Decoding *only the necessary slices* may approximately double the average decode rate.

EX-1003 ¶[0111].

Because the first D (refresh depth) rows in the second picture (e.g., picture 152) are above the I-slice, and the second D rows in the second picture are the I-slices, decoding the “(2 x refresh depth) rows” decodes the rows above the I-slices and the I-slices themselves, but the rows below the I-slices need not be decoded. *See* EX-1003 ¶¶[0095], [0068], [0111]. Similarly, because in picture 154 “the first 2*D rows are not I-slices, and the 3rd D rows are [I-slices],” the rows above the I-slices and the I-slices are decoded, but the rows below the I-slices need not be decoded. *Id.* ¶¶[0095], [0111], Fig. 4A:



Thus, because the rows above the I-slices and the I-slices in each picture may be decoded, and because *MacInnis* also discloses omitting decoding the slices below the I-slices, *MacInnis* discloses “decoding at least a portion of another picture...without having decoding the at least one slice below the intracoded slice.” EX-1002 ¶ 124.

2. Claim 2

- a. Element [2.1]: *MacInnis* discloses “wherein the another picture comprises an intracoded slice and at least one slice above the intracoded slice.”**

MacInnis discloses element [2.1]. As discussed above with respect to elements [1.4]-[1.7], *MacInnis* explains that decoding starts with a picture whose first row is an I-slice, and may continue successively “until a picture is decoded where the bottom row is an I-slice.” ¶[0060].

For example, after decoding pictures 150 and 152, decoder 120 may decode picture 154 (the claimed “another picture”), which is described as having the “first 2*D rows” that “are not I-slices, and the 3rd D rows are.” *See, e.g., id.* ¶[0056]-[0060].

Picture 154 (the claimed “another picture”) comprises at least one I-slice (the claimed “an intracoded slice”) and 2*D rows that are not I-slices (the claimed “at least one slice above the intracoded slice”). Thus, *MacInnis* teaches element [2.1].

- a. Element [2.2]: *MacInnis* discloses “wherein decoding at least a portion of the another picture further comprises: decoding the at least one slice above the intracoded slice in the second picture; and decoding the intracoded slice of the second picture.”**

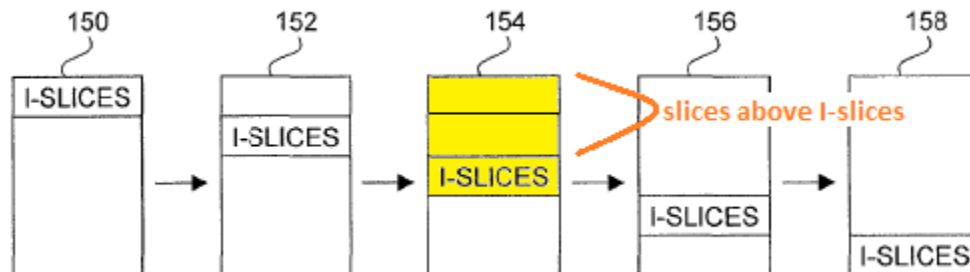
MacInnis discloses element [2.2], which is broken into subparts below due to its length.

- i. Element [2.2.a]: *MacInnis* discloses “wherein**

decoding at least a portion of the another picture further comprises: decoding the at least one slice above the intracoded slice in the second picture.”³

See discussion of elements [1.5] and [2.1] above. *MacInnis* explains that decoding starts with a picture whose first row is an I-slice, and may continue successively “until a picture is decoded where the bottom row is an I-slice.” *Id.* ¶¶[0060], [0058].

For picture 154, the first D, second D, and third D rows “at the top [of the picture] preferably are decoded.” See, e.g., *id.* ¶[0095] (explaining that rows at the top of each picture, including the I-slices, are decoded) and Fig. 4A:

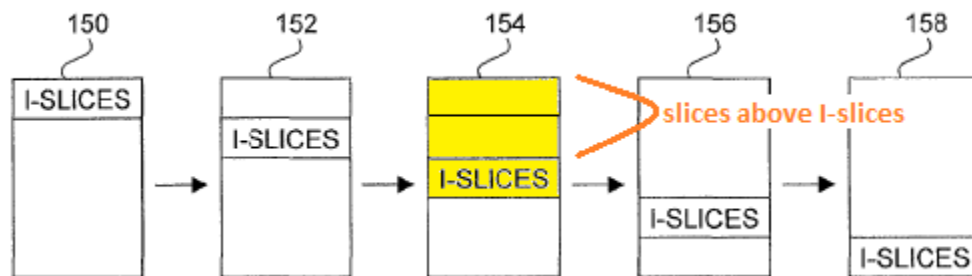


Because the first D and second D rows of picture 154 are decoded, and the first D and second D rows include slices above the I-slices, *MacInnis* discloses the claimed “decoding the at least one slice above the intracoded slice in the second picture.” EX-1002 ¶130.

³ Claim 2 switches between discussing the “another picture” and the “second picture.”

- ii. **Element [2.2.b]: *MacInnis* discloses “wherein decoding at least a portion of the another picture further comprises: . . . decoding the intracoded slice of the second picture.”**

See discussion of elements [1.6], [2.1], and [2.2.a], above. MPEG video decoder 120 may decode the first D, second D, and third D rows of picture 154, where the first D and second D rows are not I-slices, but the 3rd D rows *are* I-slices. *See* EX-1003 ¶[0057]. *See again* ¶[0095] and Fig. 4A:



Because the third D rows of picture 154 are decoded, and the third D rows include I-slices, *MacInnis* discloses the claimed “decoding the intracoded slice of the second picture.” EX-1002 ¶132.

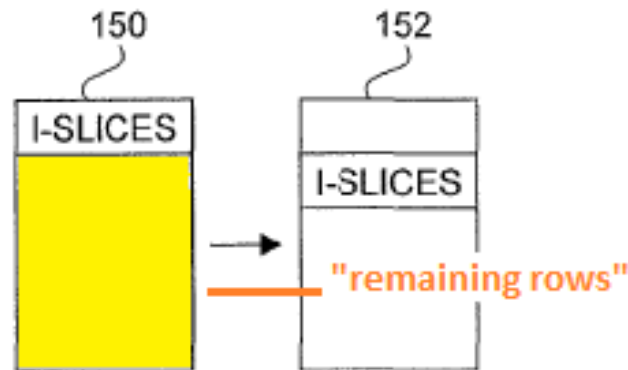
3. **Claim 3: *MacInnis* discloses “wherein causing decoding the at least one slice above the intracoded slice in the another picture comprises causing: predicting the slices above the intracoded slice in the another picture from the intracoded slice and the slices above the intracoded slice in the picture.”**

MacInnis discloses claim 3. Each of the pictures in the stream is one of a P-picture (predicted picture) or a B-picture (bi-directionally predicted picture). *See*

EX-1003 ¶[0053]. For example, in picture 150, other than the I-slices at the top of the picture,

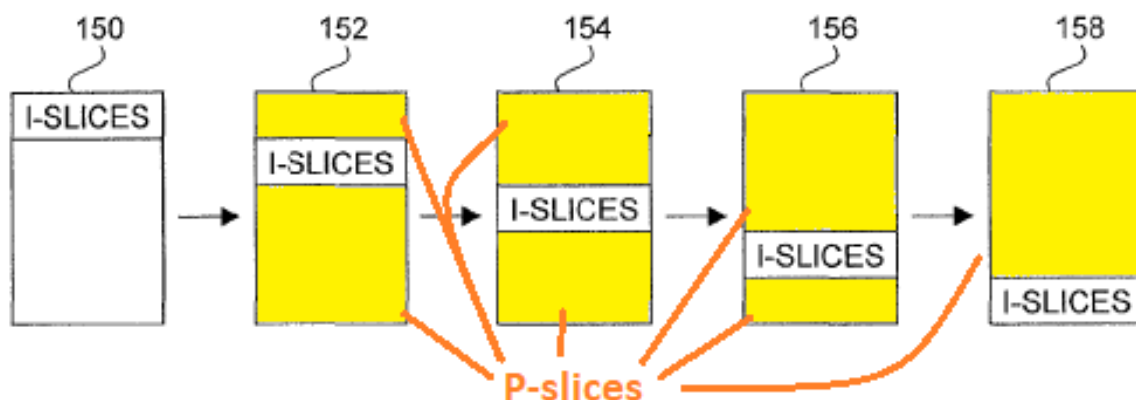
[t]he *remaining rows* typically are predictively coded, i.e. P-slices, even though it is possible for *the remaining rows* to include I-slices in case all macroblocks happen to be intra coded.

See also Fig. 4A:



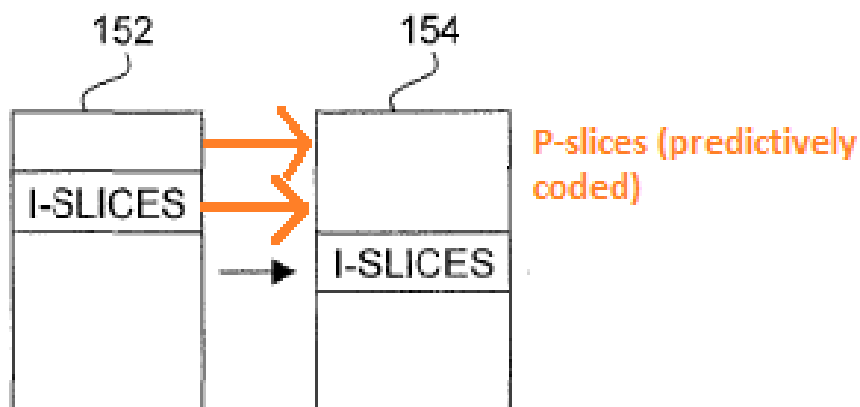
See also ¶[0058] (explaining that motion vectors in pictures where the I-slices are “not at the top of the picture” are limited to pointing at rows at or above their I-slices).

The rows above the I-slices in each picture may also be P-slices. EX-1002 ¶135. As the specification explains, for example, in picture 152, “the first D rows are not I-slices.” EX-1003 ¶[0057]. A POSITA would understand from this disclosure that the slices in all “remaining rows” in pictures 152, 154, 156, and 158 may also be predictively coded (P-slices). See *again* Fig. 4A:



Because the slices above the I-slices in picture 154 are predictively coded, those slices are predicted using the I-slices and P-slices above the I-slices in picture 152.

See again Fig. 4A:



See also ¶[0058] and EX-1002 ¶136. Thus, because *MacInnis* teaches predicting the slices above the intracoded slice in picture 154 (claimed “predicting the slices above the intracoded slice in the another picture”) using the I-slices and slices

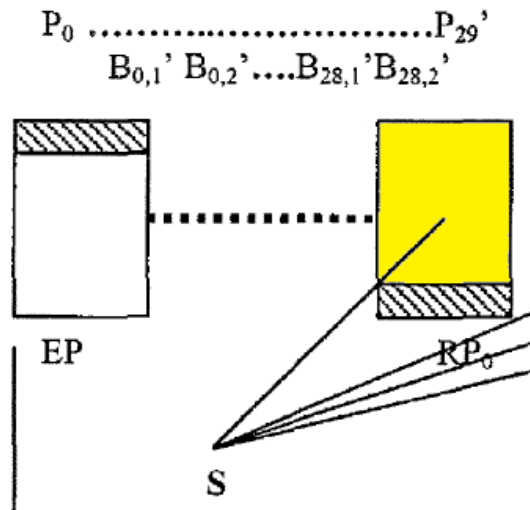
above the I-slices in picture 152 (“predicting...from the intracoded slice and the slices above the intracoded slice in the picture”), *MacInnis* discloses claim 3.

4. Claim 4: *MacInnis* discloses “wherein decoding the at least one slice above the intracoded slice further comprises causing: decoding the at least one slice in raster order.”

The '967 patent discloses a top-to-bottom order of decoding for “raster order.” *See* EX-1001 6:46-52:

Accordingly, aspects of the present invention involve decoding the slices, S, of the HITS stream P-pictures, e.g., $P_0' \dots P_{28}'$ which are above the last intracoded slice I in the picture. If the slices, S, I of the P-pictures, $P_0' \dots P_{28}'$, are decoded in raster order, the video decoder 345 decodes *each slice of the P-picture, $P_0' \dots P_{28}'$, until the last intracoded slice, I* is decoded.

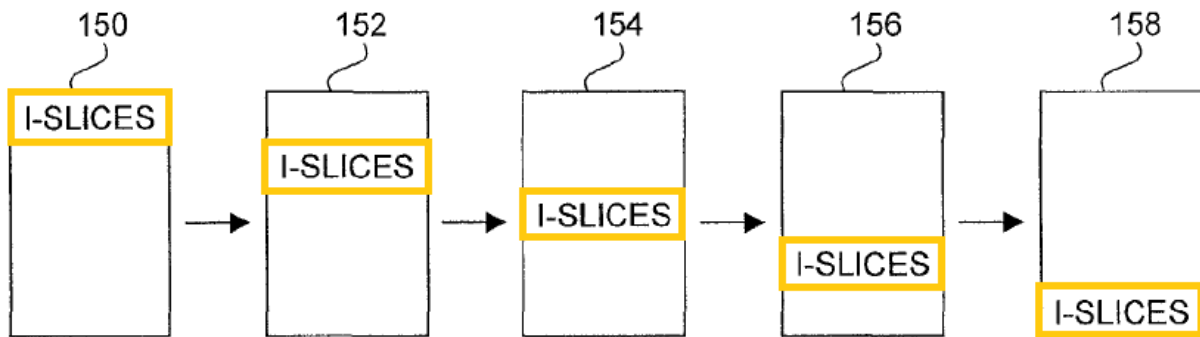
Fig. 4 depicts this top-to-bottom decoding order which decodes the slices (“S”) “which are above the last intracoded slice I in the picture” from top to bottom (highlighted in yellow):



Id. 6:48-49.

MacInnis similarly teaches decoding I-slices from top to bottom. For example, decoding may start at picture 150, where “the first D rows...are I-slices.”

Id. ¶[0056]. Decoding may continue to picture 152, where “the first D rows are not I-slices, and the second D rows are” I-slices. *Id.* ¶[0057]. Decoding may then continue to picture 154 (the claimed “another picture”), similarly described as having the “first 2*D rows” that “are not I-slices, and the 3rd D rows are.” *See, e.g., id.* ¶¶[0056]-[0060], Fig. 4A:



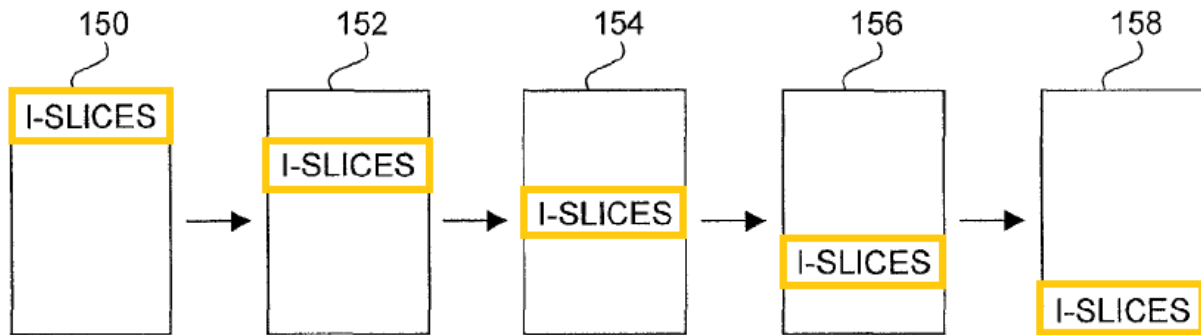
MacInnis thus teaches decoding a top-most slice (in picture 150), followed by slices that progressively move from top-to-bottom, until decoding a bottom-most slice (in picture 158). EX-1002 ¶¶140-141.

Accordingly, because *MacInnis* teaches decoding the first D rows in a first picture, followed by the first D and second D rows in a second picture, followed by the first D, second D, and third D rows in a third picture, etc., *MacInnis* teaches “decoding the at least one slice in raster order” as required by claim 4.

5. Claim 5: *MacInnis* discloses “wherein the picture comprises a P-picture from a HITS stream.”

MacInnis discloses claim 5. *MacInnis* relates to decoding digital streams, such as HITS streams, that do not include I-pictures but do include P-pictures (predicted picture) or B-pictures (bi-directionally predicted picture). EX-1003 ¶¶[0008], [0050], [0053].

The stream depicted in Figure 4A includes I-slices that “follow a consistent pattern” in that they “appear sequentially from top to bottom” from start to finish in the stream. EX-1003 ¶[0056]; Fig. 4A:



MacInnis explains that this pattern of I-slices only applies to P pictures.

¶[0056]. Thus, because picture 152 (the claimed “picture”) is a P-picture from a HITS stream, *MacInnis* teaches claim 5.

B. Ground 2: The combination of *MacInnis* and ISO 13818-2 renders claim 4 obvious.

- 1. Claim 4: *MacInnis* in view of ISO 13818-2 renders obvious “wherein decoding the at least one slice above the intracoded slice further comprises causing: decoding the at least one slice in raster order.”**

As discussed above, *MacInnis* teaches a top-to-bottom decoding order for the slices in a bitstream. *See, e.g.*, EX-1003 Fig. 4A, ¶¶[0056]-[0060]. Because decoder 120 decodes the slices in the bitstream from top-to-bottom with respect to the bitstream, the claimed “at least one slice” (i.e., slices across the bitstream) is decoded “in raster order.” However, to the extent that Broadcom argues or the

Board finds, before or after institution, that *MacInnis* does not expressly disclose “decoding the at least one slice in raster order,” *ISO 13818-2* expressly discloses this step.

MacInnis specifically refers to and builds upon the MPEG-2 standard in *ISO 13818-2*. ¶[0054] (“Rows and slices [in *MacInnis*] are defined in the MPEG-2 video standard, *ISO/IEC 13818-2*”); see also, e.g., ¶¶[0026], [0033], [0037], and [0050] (referring to “MPEG-2 video streams”). A POSITA would have thus started with *ISO 13818-2* when implementing the teachings of *MacInnis*. EX-1002 ¶148.

ISO 13818-2, which the '967 patent admits dominated distribution of digital video content for TV display at the time the application that matured into the '967 patent was filed, EX-1001 1:31-33, explains that a “slice” is a “series of an arbitrary number of consecutive macroblocks.” EX-1009 18 (§ 6.1.2). Each slice in a frame “shall occur in the bitstream in the order in which they are encountered, starting at the upper-left of the picture and *proceeding by raster-scan order* from left to right and top to bottom (illustrated in Figures 6-8 and 6-9 as alphabetical order).” *Id.* Figure 6-9 on page 19 depicts this top to bottom order of encoding and decoding video data as being in alphabetical order:

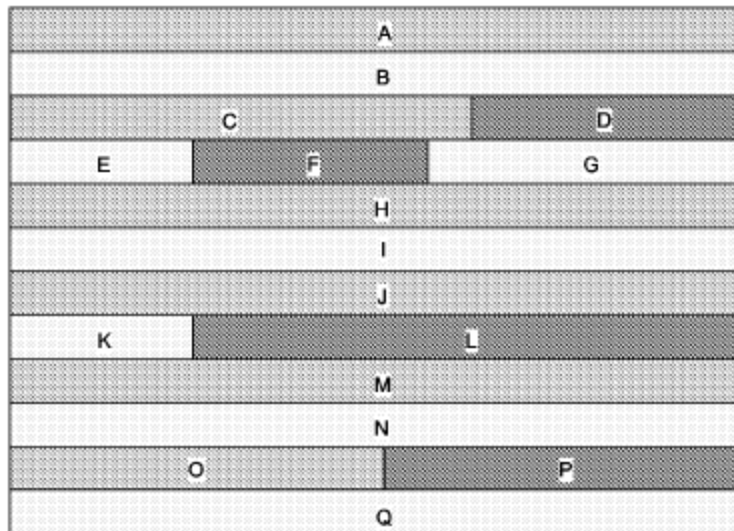


Figure 6-9 – Restricted slice structure

Having started with *ISO 13818-2* when implementing the teachings of *MacInnis*, based on these disclosures of *ISO 13818-2*, a POSITA would have found it obvious to implement the teachings of *MacInnis* so as to decode at least one slice in raster order. Such implementation would have been straightforward. EX-1002 ¶150. Because *ISO 13818-2* explains that MPEG data (such as in *MacInnis*) is encoded in raster order, decoding in raster order would be the most straightforward method of decoding. EX-1002 ¶150. Raster order – e.g., “from top to bottom” – is also a natural method of decoding data because it is a straightforward application of “first-in, first-out” (FIFO) principles. EX-1002 ¶150; *see also* EX-1010 919 (defining raster order as “the order in which pixels are scanned in a raster display device; usually left-to-right; top-to-bottom”).

A POSITA would have considered decoding slices in raster order obvious in view of the combined disclosures of *MacInnis* and *ISO 13818-2*. EX-1002 ¶151. Applying the known techniques of *ISO 13838-2* in *MacInnis* would have yielded predictable results and known advantages. Accordingly, a POSITA would have found it obvious to implement decoder 120 in *MacInnis* so as to “decod[e] the at least one slice in raster order,” as required by claim 4. EX-1002 ¶151.

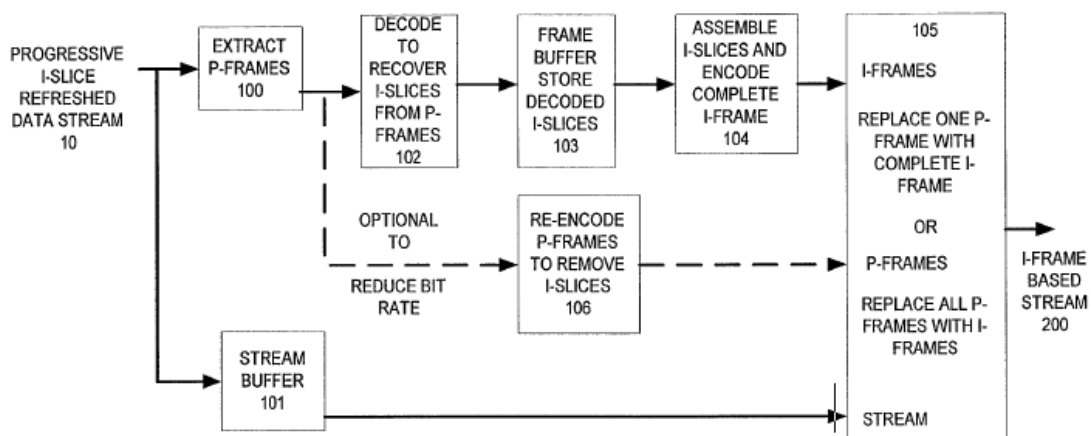
C. Ground 3: *Chen* ‘677 anticipates claims 1-4

U.S. Patent Application Publication No. 2001/0026677 to Chen et al. (“*Chen* ‘677”) published on October 4, 2001. EX-1004. *Chen* ‘677 is prior art under at least 35 U.S.C. § 102(b). *Supra* at Section II.D. To the extent that Broadcom asserts or the Board finds, before or after institution, that the challenged claims of the ‘967 patent are entitled to the filing date of the CIP parent application (which they are not), *Chen* ‘677 would still be prior art under at least § 102(e) but is not co-owned and thus cannot be excluded under § 103(c).

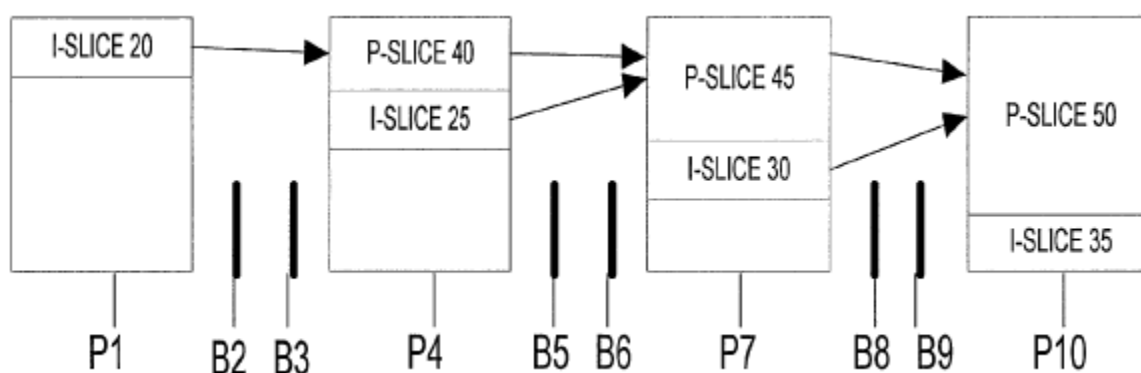
Like the ‘967 patent, *Chen* ‘677 relates to decoding “progressive I-slice refreshed MPEG data streams.” EX-1004 ¶[0003]; *see also* EX-1001 2:6-16. The phrase “progressive refresh” (as used in *Chen* ‘677) refers to the manner of decoding a picture. *See* EX-1004 ¶¶[0005], [0009]. For example, in a progressive refresh scheme, a decodable portion (such as an I-slice) in each picture in a set of pictures moves from the top of one picture to the bottom of the next. EX-1004 Fig.

1, ¶¶[0022]-[0025]. A P-picture can only be decoded (“refreshed”) once a set of P-pictures has been decoded such that each portion of that original P-picture has been decoded. EX-1004 ¶¶[0022], [0031]; *see* EX-1007 1:22-25 (explaining refreshing of P-pictures).

Chen ‘677 recognizes the difficulty in implementing “trick play modes in a progressive I-slice refreshed MPEG data stream,” stating that “the lack of I-frames requires either extremely large buffers for several pictures worth of data or the stream must be fully decoded and then re-encoded as an I-frame based stream.” ¶[0005]. The process in *Chen* ‘677 solves this problem by “transcoding progressive I-slice refreshed MPEG data streams to I-frame based MPEG data streams to enable trick play modes on a television appliance.” *Id.* ¶[0021]. Figure 2 represents this process of transcoding from a progressive stream to a stream with I-frames.



Television appliance 300 receives a progressive I-slice refreshed stream (10) in step 100 and extracts the P-frames. EX-1004 ¶[0025]. The stream is buffered in step 101, and the P-frames are decoded “to recover the I-slices” which are distributed over multiple P-frames and “make up a complete I-frame.” *Id.* ¶[0012], [0025]. Fig. 1 shows an example of such a stream, where the I-frame is divided into slices and spread across P-frames:



Chen '677 explains that “[t]hose skilled in the art will appreciate that the P-frames need only be *partially decoded* in order to recover the I-slices.” *Id.* ¶[0025].

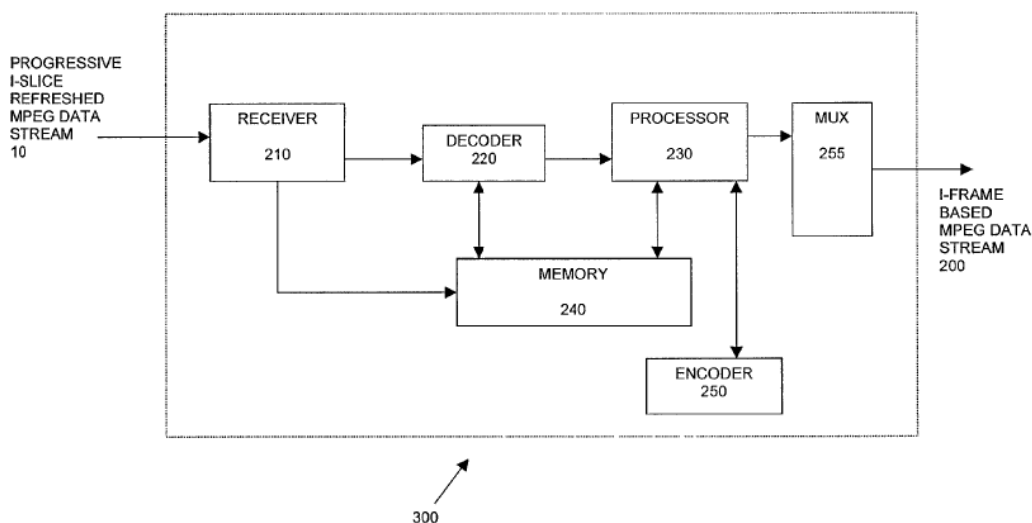
Reconstructing a P-frame requires decoding at least the previous frame. *Id.* ¶[0022]. For example, I-slice 25 and P-slice 40 in P-frame P4 provide “reference macroblocks that can be used to predict” the contents of P-slice 45 in P-frame P7. *Id.* ¶[0023]. Macroblocks are square-shaped blocks representing picture data. EX-1009 5 (definition 3.89); EX-1002 ¶74.

As further detailed below, *Chen '677* anticipates claims 1-4 under constructions offered by Broadcom in the related ITC Investigation. EX-1008.

1. Claim 1

a. Element [1.0]: *Chen '677* discloses “[a] system for displaying pictures.”

To the extent that Broadcom argues or the Board finds, before or after institution, that the preamble limits the claims, *Chen '677* discloses it. *Chen '677* discloses a television appliance 300, depicted in Fig. 3:



See also EX-1004 ¶¶[0027] and [0035]. Television appliance 300 includes a receiver 210 that receives an MPEG data stream 10. *Id.* ¶[0027], Fig. 3. The receiver 210 sends the stream to decoder 220 which decodes P-frames in the stream to recover I-slices (intra-coded slices) “which make up a complete I-frame.” *Id.* ¶[0027], Fig. 3. Memory 240 may additionally or alternatively buffer I-slices

while decoding the P-frames and may buffer the MPEG data stream 10. ¶[0028], Fig. 3. Processor 230 reassembles the slices into a complete I-frame and encoder 250 encodes the complete I-frame. ¶[0027], Fig. 3. Multiplexer 255 replaces P-frames in stream 10 with an encoded I-frame in order to provide an encoded I-frame based stream 200. ¶[0027], Fig. 3. This stream can be stored for “trick play mode use.” ¶[0029].

Thus, because *Chen* ‘677 discloses a system that receives a stream with P-frames, extracts I-slices from the stream, reassembles the stream using I-frames, and stores the stream for playback, *Chen* ‘677 discloses element [1.0]. EX-1002 ¶154.

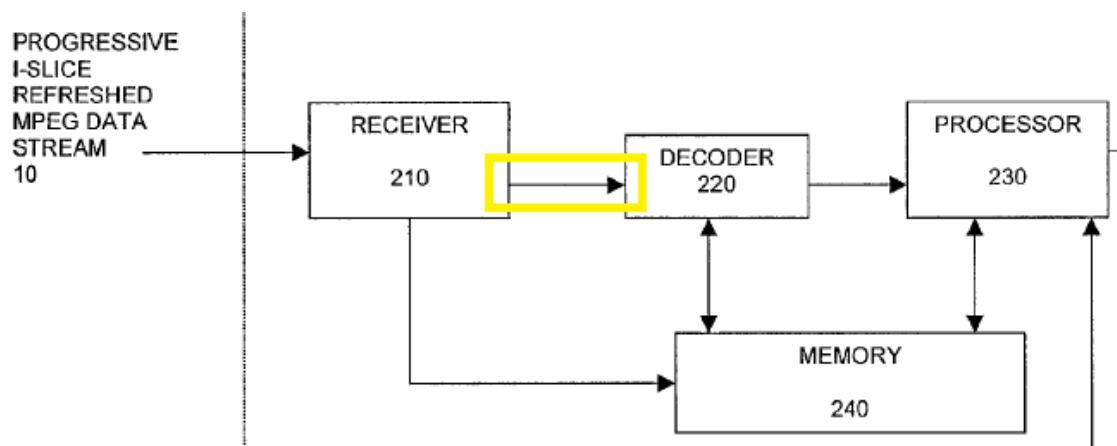
b. Element [1.1]: *Chen* ‘677 discloses “a host processor for transmitting transport packets, said transport packets providing a plurality of instructions.”

As discussed above, Broadcom has asserted in related litigation, ITC Investigation No. 337-TA-1047, that the phrase “host processor” means

any device or element that processes an incoming transport stream (i.e., a stream of packets), extracts the packets from the stream, and passes the packets on to a decoder device (such as any audio or visual decoding device or functionality),

citing 3:45-52 of the ‘967 patent. EX-1008. Under at least this construction, *Chen* ‘677 teaches the claimed “host processor.”

For example, *Chen* '677 depicts receiver 210 receiving a progressive I-slice refreshed data stream 10. EX-1004 Fig. 2, ¶¶[0025]-[0027] (“progressive I-slice refreshed MPEG data stream 10 having I-slices distributed over multiple P-frames is received by a television appliance 300 (e.g., by receiver 210)”). The stream is transmitted from the receiver 210 to the decoder 220, where the P-frames of the stream are decoded. ¶[0027]. The arrow between receiver 210 and decoder 220 in Fig. 3 indicates that the stream is transmitted from receiver 210 to decoder 220 for decoding:



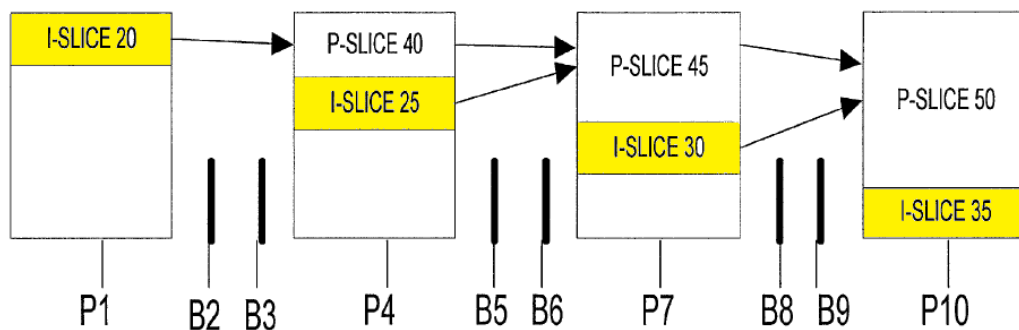
Thus, *Chen* '677 teaches the claimed “host processor.”

Broadcom also asserts that the claimed “plurality of instructions” is “the contents of the coded or compressed audio or video syntax itself, which instruct or cause the decoder to perform various tasks.” EX-1008 3. To the extent that Broadcom asserts or the Board finds, before or after institution, that Broadcom’s

construction of “plurality of instructions” from the ITC proceeding is the broadest reasonable construction, *Chen* ‘677 teaches this element.

For example, as discussed above, *Chen* ‘677 depicts receiver 210 receiving a progressive I-slice refreshed data stream 10. EX-1004 Fig. 2, ¶¶[0025]-[0027] (“stream 10 . . . is received by a television appliance 300 (e.g., by receiver 210)”). The stream comprises a series of I-slices distributed over multiple P-frames. *Id.*

¶[0025], Fig. 1:



Chen ‘677 explains that this “progressive I-slice refreshed MPEG data stream” is a type of “compressed MPEG data stream.” *Id.* ¶[0009]. The stream is transmitted from the receiver 210 to the decoder 220. ¶[0027]. The stream causes the decoder to decode it to recover the I-slices. ¶[0027], *see also* Fig. 2, step 102.

Thus, because *Chen* ‘677 discloses receiver 210 transmitting a compressed data stream to a decoder 220, and further teaches that the stream causes decoder 220 to decode it, *Chen* ‘677 discloses the claimed “plurality of instructions” under at least Broadcom’s construction (“the contents of the coded or compressed audio

or video syntax itself, which instruct or cause the decoder to perform various tasks”). EX-1008 3; EX-1002 ¶¶158-61.

As discussed above, *supra at* Section IV.C, the term “transport packet” means a packet containing either data or instructions. Because the meaning of “transport packet” includes the term “instructions,” Broadcom’s construction, *supra at* Section V.A.1.b, for “transport packet” is “a packet containing either data or the contents of the coded or compressed audio or video syntax itself, which instructs or causes the decoder to perform various tasks.”

While *Chen* ‘677 does not explicitly explain that the MPEG data stream 10 comprises “transport packets,” MPEG streams, such as stream 10 in *Chen* ‘677, comprise transport packets. *Chen* ‘677 explains that the stream is an MPEG stream. See EX-1004 ¶¶[0002]-[0012] (referring to stream 10 as being an “MPEG” stream), Fig. 3 (“Progressive I-slice Refreshed *MPEG* Data Stream 10”). Such MPEG streams comprised a Packetized Elementary Stream (“PES”) that was multiplexed into a Transport Stream (TS) at the time of filing of *Chen* ‘677. The ISO 13818-2 standard, which the ‘967 patent recognized as dominating distribution of digital video content for TV display, 1:30-33, published in 2000. EX-1009 2. It explains that MPEG video may be stored and sent in the form of a PES multiplexed into a TS. See, e.g., *id.* 154 (referring to storing information for

decoding the bitstream in the “PES packet”), 155 (same); *see also* EX-1011 647, defining “PES” as

Packetized Elementary Stream: In MPEG-2, after the media stream has been digitized and compressed, it is *formatted into packets* before it is multiplexed into either a Program Stream or *Transport Stream*.

Such packets must be extracted from the received MPEG data stream 10 (e.g., a Transport Stream) before they can be decoded. *See, e.g.*, EX-1012 18 (“Transport Streams are . . . unpacketized by a DePacketizer;” see Fig. 2.1); EX-1002 ¶164; EX-1013 ¶ 3, EX-1014. Because the packets were multiplexed into stream 10, they are also demultiplexed for retrieval and forwarding to decoder 220. *See, e.g.*, EX-1012 18 (“Transport Streams are decoded by a Transport Demultiplexer;” see Fig. 2.1); EX-1002 ¶164.

Because MPEG streams are stored and sent in the form of packetized streams, and because *Chen* ’677 teaches a receiver 210 that receives an MPEG stream 10 and extracts data in the form of packets from the MPEG stream, and, *Chen* ’677 teaches that the “plurality of instructions” transmitted by a “host processor” is stored and sent in a series of “transport packets.”

Therefore, *Chen* ’677 discloses element [1.1].

c. Element [1.2]: *Chen* ‘677 discloses “a video decoder for executing the plurality of instructions.”

See above discussion of element [1.1]. *Chen* ‘677 discloses element [1.2].

For example, as discussed above with respect to element [1.1], *Chen* ‘677 depicts receiver 210 receiving a progressive I-slice refreshed data stream 10. EX-1004 Fig. 2, ¶¶[0025]-[0027]. Starting in step 100, a progressive I-slice refreshed stream (10) is received and the P-frames are extracted. ¶[0025]. Step 102 represents decoder 220 decoding the P-frames in the stream “to recover the I-slices” which are distributed over multiple P-frames and “which make up a complete I-frame.” *See id.* ¶¶[0025], [0027] (discussing decoder 220).

Thus, because *Chen* ‘677 discloses a decoder 220 that decodes stream 10 to recover I-slices, *Chen* ‘677 discloses element [1.2]. EX-1002 ¶¶167-68.

d. Element [1.3]: *Chen* ‘677 discloses “wherein execution of the instructions by the video decoder comprises causing”

See above discussion of element [1.1]. As discussed above with respect to element [1.2], decoder 220 decodes the P-frames in the stream “to recover the I-slices” which are distributed over multiple P-frames and “which make up a complete I-frame.” *See id.* ¶¶[0025], [0027] (discussing decoder 220). Decoder 220 “executes” the stream in that the stream causes decoder 220 to extract the I-slices. EX-1002 ¶169; *supra* discussion of element [1.1].

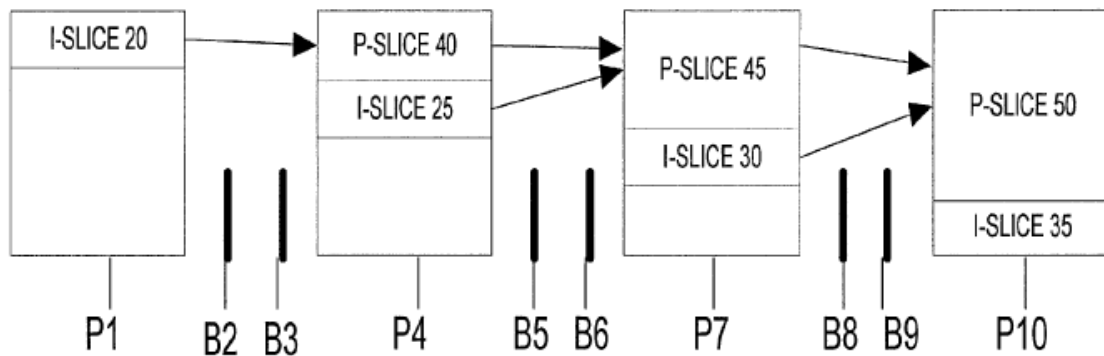
Because decoder 220 receives an MPEG stream, and executes the data in the stream to operate on the stream (e.g., to extract I-slices), decoder 220 in *Chen '677* teaches executing “the instructions by the video decoder.”

- e. **Element [1.4]: *Chen '677* discloses “selecting a picture, said picture comprising an intracoded slice and at least one slice above the intracoded slice, and at least one slice below the intracoded slice.”**

Chen '677 discloses element [1.4], which is broken into subparts below due to its length.

- i. **Element [1.4.a]: *Chen '677* discloses “selecting a picture, said picture comprising an intracoded slice and at least one slice above the intracoded slice.”**

Decoder 220 in *Chen '677* decodes “P-frames to recover the I-slices which make up [a] complete I-frame.” ¶[0030], *see also* Fig. 1. For example, in Figure 1, ten frames are depicted – P-frames P1, P4, P7, and P10, and B-frames B2, B3, B5, B6, B8, and B9.



See also id. ¶[0023]. Decoding all 10 frames is “required before a picture is completely refreshed [(i.e., decoded)] using all four I-slices 20, 25, 30, and 35.” *Id.*

For example, decoder 220 may decode I-slice 20 in frame P1. EX-1004 ¶[0023]. I-slice 20 “provides reference macroblocks used to create motion vectors that predict the contents of P-slice 40 in P-frame P4.” *Id. see also* Fig. 1. After decoding P-frame P1, decoder 220 may decode P-frame P4. *See id.* ¶[0023], Fig. 1. P-frame P4 may include I-slice 25 and P-slice 40. *See, e.g.,* ¶[0023], Fig. 1. Decoder 220 may select each frame in the stream and determine its type (e.g., I picture, B picture, P picture). *See id.* ¶[0037], Fig. 4 (steps 300-325). P-frames are decoded to recover I-slices, frame by frame, “until a complete refresh cycle has passed.” *See id.* ¶[0040], Fig. 4 (steps 355-370). *Chen* ’677 thus teaches “selecting a picture.” EX-1002 ¶174.

Because P-frame P4 contains an intracoded slice, I-slice 25, and at least one slice above the intracoded slice, P-slice 40, P-frame P4 constitutes the claimed “picture comprising an intracoded slice and at least one slice above the intracoded slice.” EX-1002 ¶175.

ii. Element [1.4.b]: *Chen* ’677 discloses “said picture comprising...at least one slice below the intracoded slice.”

Figure 1 of *Chen* ’677 depicts P-frame P4 (claimed “said picture”), and depicts that P4 contains an I-slice 25 and a predictively coded P-slice 40 above

I-slice 25. *See, e.g.*, EX-1004 ¶[0023], Fig. 1. *Chen* '677 also explains that “typically,” each frame, such as P-frame P4, “will be a ‘full’ video frame.” ¶[0020]. Because P-frame P4 is described as a “full” frame, P-frame P4 comprises additional slices below depicted I-slice 25. EX-1002 ¶176.

As discussed above with respect to element [1.1], stream 10 in *Chen* '677 is an MPEG stream. *See* EX-1004 ¶¶[0002]-[0012], Fig. 3 (“Progressive I-slice Refreshed *MPEG* Data Stream 10”); EX-1009 154, 155 (same); *see also* EX-1011 647.

The MPEG standard (ISO 13818-2) shows that slices in MPEG streams may encompass a “full” frame. EX-1009 19, Figure 6-9:

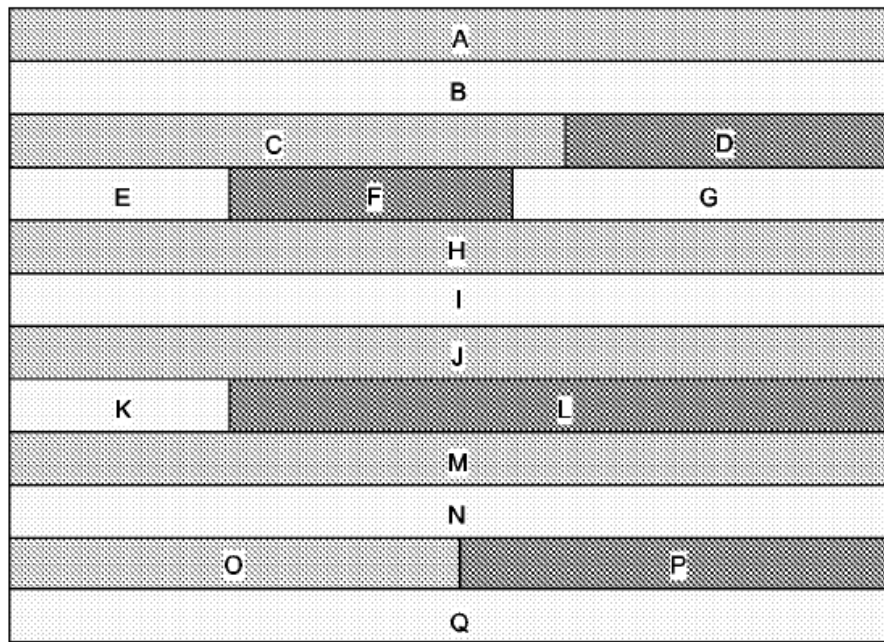


Figure 6-9 – Restricted slice structure

Because *Chen* '677 teaches that frames may be “full” and MPEG pictures may comprise slices that encompass an entire picture, *Chen* '677 teaches element [1.4.b].

f. Element [1.5]: *Chen* '677 discloses “decoding the at least one slice above the intracoded slice.”

Chen '677 discloses element [1.5]. “I-slice 25 and P-slice 40 contained in P-frame P4 provide reference macroblocks that can be used to predict the contents of P-slice 45 contained in P-frame P7.” EX-1004 ¶[0023]. This means that in order to eventually decode P7, P-slice 40 and I-slice 25 in P-frame P4 must also be decoded. *See id.* ¶[0023]. ¶[0040] also explains that recovering I-slices requires the

decoding of P-frames, which includes P-slices (such as P-slice 40). EX-1004 ¶[0040].

As discussed above, decoder 220 may decode part of P-frame P4 (the claimed “picture”). *See* EX-1004 ¶[0023], Fig. 1. Decoding P-frame P4 includes decoding P-slice 40 above I-slice 25. *See, e.g., id.* ¶[0023], Fig. 1. Because P-frame P4 contains an intracoded slice, I-slice 25, and at least one slice above the intracoded slice, P-slice 40, and decoder 220 decodes P-slice 40, *Chen ‘677* discloses “decoding the at least one slice above the intracoded slice.” EX-1002 ¶181.

g. Element [1.6]: *Chen ‘677* discloses “decoding the intracoded slice.”

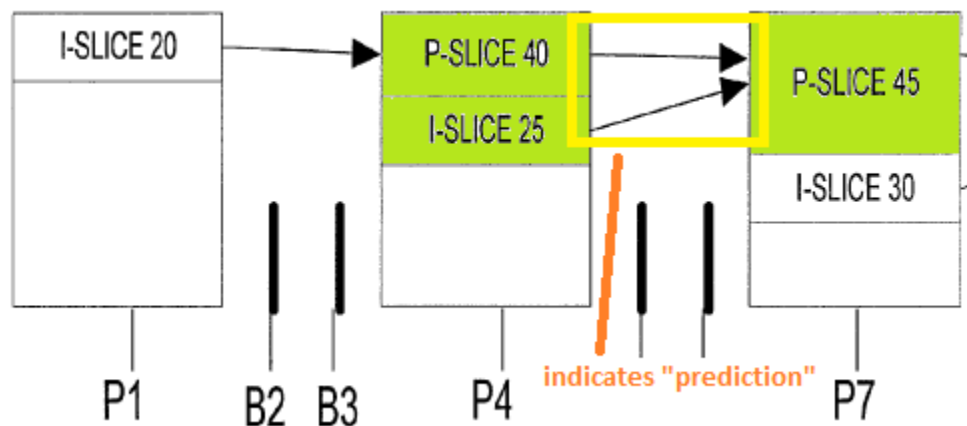
See above discussion of element [1.5]. “I-slice 25 and P-slice 40 contained in P-frame P4 provide reference macroblocks that can be used to predict the contents of P-slice 45 contained in P-frame P7.” EX-1004 ¶[0023].

As discussed above, decoder 220 may decode part of P-frame P4, which includes decoding I-slice 25. *See, e.g., id.* ¶[0023], Fig. 1. Because decoding P-frame P4 decodes intracoded slice I-slice 25, *Chen ‘677* discloses “decoding the intracoded slice.” EX-1002 ¶¶182-83.

h. Element [1.7]: *Chen ‘677* discloses “decoding at least a portion of another picture after decoding the at least one slice above the intracode slice and the intracode

slice without having decoding the at least one slice below the intracoded slice.”

Chen ‘677 discloses element [1.7]. *Chen* ‘677 explains that each P-frame “contains an abridged set of data used by the decoder to predict a full frame from a previous ‘P’ frame or from a previous complete frame...in the video stream.” EX-1004 ¶[0003]. As discussed above, decoder 220 may decode part of P-frame P4, such as P-slice 40 and I-slice 25. *See* EX-1004 ¶[0023], Fig. 1. After decoding P-frame P4, decoder 220 may decode part of P-frame P7, such as P-slice 45 and I-slice 30. *See id.* ¶[0023], Fig. 1. For example, decoder 220 may utilize the information decoded from P-frame P4, such as “reference macroblocks” from P-slice 40 and I-slice 25 to “predict the contents” of P-slice 45 in P-frame P7. *See id.* ¶[0023], Fig. 1:



Decoding at least part of P-frame P7, after decoding P-slice 40 and I-slice 25 in P-frame P4, thus discloses “decoding at least a portion of another picture after

decoding the at least one slice above the intracode slice and the intracode slice” as required by element [1.7].

Chen '677 extracts the P-frames from MPEG stream 10 by copying the P-frames and/or buffering the stream. *Id.* ¶[0025]. *Chen '677* explains that “[t]hose skilled in the art will appreciate that the *P-frames need only be partially decoded* in order to recover the I-slices, although the entire P-frame may be decoded to recover the I-slices if desired.” *Id.* As depicted in Figure 1, the I-slices in each frame are stored in descending order. ¶[0023] explains that each of the I-slices in P-frames P1, P4, P7, and P10 are required to refresh a full picture.

Because the I-slices required to decode and reconstruct a complete picture are obtained by decoding P-frames P1, P4, P7, and P10 to obtain the I-slices, and because *Chen '677* explains that an I-slice in a P-frame may be decoded without decoding the rest of the respective P-frame, *Chen '677* discloses the claimed “decoding at least a portion of another picture...without having decoding the at least one slice below the intracoded slice.” EX-1002 ¶187.

2. Claim 2

- a. **Element [2.1]: *Chen* ‘677 discloses “wherein the another picture comprises an intracoded slice and at least one slice above the intracoded slice.”**

Chen ‘677 discloses element [2.1]. As discussed above with respect to elements [1.4]-[1.7], and as depicted in Figure 1, P-frame P7 comprises an intracoded slice (I-slice 30) and a slice above I-slice 30 (P-slice 45).

Because P-frame P7 contains an intracoded slice and at least one slice above the intracoded slice, P-frame P7 in *Chen* ‘677 discloses “the another picture comprises an intracoded slice and at least one slice above the intracoded slice,” as required by element [2.1]. EX-1002 ¶189.

- b. **Element [2.2]: *Chen* ‘677 discloses “wherein decoding at least a portion of the another picture further comprises: decoding the at least one slice above the intracoded slice in the second picture; and decoding the intracoded slice of the second picture.”**

Chen ‘677 discloses element [2.2], which is broken into subparts below due to its length.

- i. **Element [2.2.a]: *Chen* ‘677 discloses “wherein decoding at least a portion of the another picture further comprises: decoding the at least one slice above the intracoded slice in the second picture.”⁴**

⁴ Claim 2 switches between discussing the “another picture” and the “second picture.”

See discussion of element [1.5] above. Decoder 220 may decode part of P-frame P7. *See* EX-1004 ¶¶[0023], [0040], Fig. 1. Decoding P-frame P7 requires decoding P-slice 45, which is above I-slice 30. *See, e.g.*, ¶[0023], Fig. 1. Because P-frame P7 contains an intracoded slice, I-slice 30, and at least one slice above the intracoded slice, P-slice 45, and decoder 220 decodes P-slice 45, *Chen* ‘677 discloses “decoding the at least one slice above the intracoded slice in the second picture.” EX-1002 ¶191.

ii. Element [2.2.b]: *Chen* ‘677 discloses “wherein decoding at least a portion of the another picture further comprises: . . . decoding the intracoded slice of the second picture.”

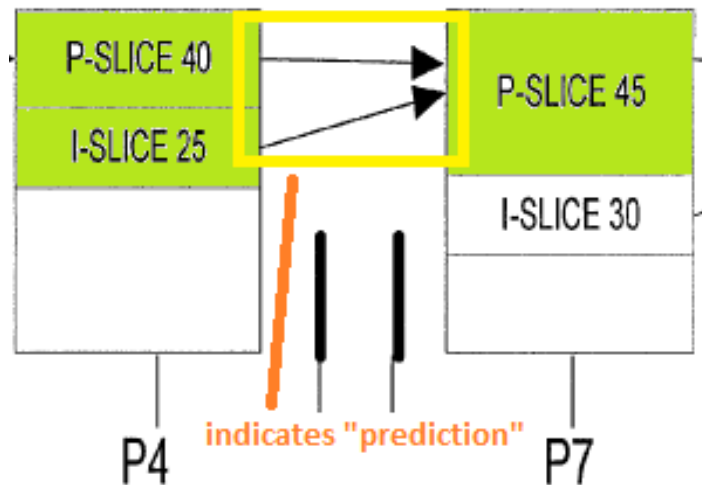
See discussion of elements [1.6] and [2.2.a] above. *Chen* ‘677 explains that “I-slice 30 and P-slice 45 contained in P-frame P7 provide reference macroblocks that can be used to predict the contents of P-slice 50 contained in P-frame P10.” ¶[0023]. Decoder 220 may decode part of P-frame P7. *See* EX-1004 ¶[0023], Fig. 1. Decoding P-frame P7 requires decoding I-slice 30. *See, e.g.*, ¶[0023], Fig. 1. Because decoding P-frame P7 decodes I-slice 30, *Chen* ‘677 discloses “decoding the intracoded slice of the second picture.” EX-1002 ¶192.

3. Claim 3: *Chen* ‘677 discloses “wherein causing decoding the at least one slice above the intracoded slice in the another picture comprises causing: predicting the slices above the intracoded slice in the another picture from the intracoded slice and the slices above the intracoded slice in the picture.”

Chen '677 discloses claim 3. *Chen '677* explains that the I-slices and/or P-slices in a first frame can be used to create motion vectors that predict the contents of slices in a following frame:

I-slice 20 contained in P-frame P1 provides reference macroblocks used to *create motion vectors* that *predict the contents of P-slice 40* in P-frame P4. In the same way, I-slice 25 and P-slice 40 contained in P-frame P4 provide reference macroblocks that can be used to *predict the contents of P-slice 45* contained in P-frame P7. Similarly, I-slice 30 and P-slice 45 contained in P-frame P7 provide reference macroblocks that can be used to *predict the contents of P-slice 50* contained in P-frame P10. I-slice 35 and P-slice 50 contained in P-frame P10 provide the next P-frame (not shown) with reference macroblocks for the predicted part of the picture.

¶[0023]. Annotated Fig. 1 below depicts “I-slice 25 and P-slice 40 contained in P-frame P4” being used to “predict the contents” of P-slice 45 in P-frame P7:



Because I-slice 25 and P-slice 40 in P-frame P4 (the claimed “intracoded slice and the slices above the intracoded slice in the picture”) are used to predict the contents of P-slice 45 which is above the intracoded slice, I-slice 30, in P-frame P7 (the claimed “predicting the slices above the intracoded slice in the another picture”), *Chen* ‘677 discloses claim 3. EX-1002 ¶¶193-94.

4. **Claim 4:** *Chen* ‘677 discloses “wherein decoding the at least one slice above the intracoded slice further comprises causing: decoding the at least one slice in raster order.”

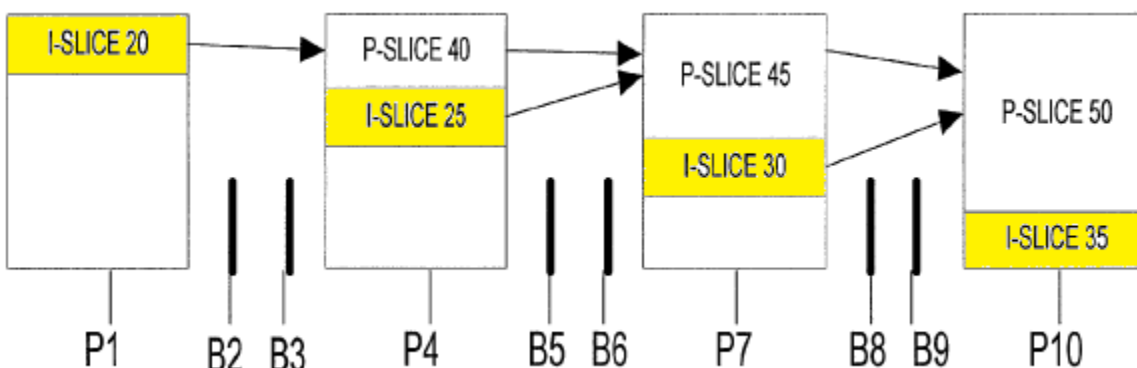
The ‘967 patent appears to refer to a top-to-bottom order of decoding for “raster order.” See EX-1001 6:46-52:

Accordingly, aspects of the present invention involve decoding the slices, S, of the HITS stream P-pictures, e.g., $P_0' \dots P_{28}'$ which are above the last intracoded slice I in the picture. If the slices, S, I of the P-pictures, $P_0' \dots P_{28}'$, are decoded in raster order, the video decoder 345

decodes *each slice of the P-picture, $P_0' \dots P_{28}'$, until the last intracoded slice, I* is decoded.

Fig. 4 depicts this top-to-bottom decoding order which decodes the slices (“S”) “which are above the last intracoded slice I in the picture” from top to bottom. *Id.* 6:48-49.

Similarly, *Chen* ‘677 teaches slice-by-slice decoding of I-slices stored in top-to-bottom order across P-frames. *See, e.g.*, EX-1004 Fig. 1:



See also id. ¶¶[0012], [0022]-[0025]. *Chen* ‘677 explains that each picture provides data (e.g., “reference macroblocks”) usable to predict a portion of a following picture. *Id.* ¶[0023]. *Chen* ‘677 thus teaches decoding a top-most slice (e.g., I-slice 20 in P-frame P1), followed by slices that progressively move from top-to-bottom, until decoding a bottom-most slice (e.g., I-slice 35 in P-frame P10). EX-1002 ¶¶197-98.

Accordingly, because *Chen* ‘677 teaches decoding the first rows in a first picture, followed by the first and second rows in a second picture, followed by the

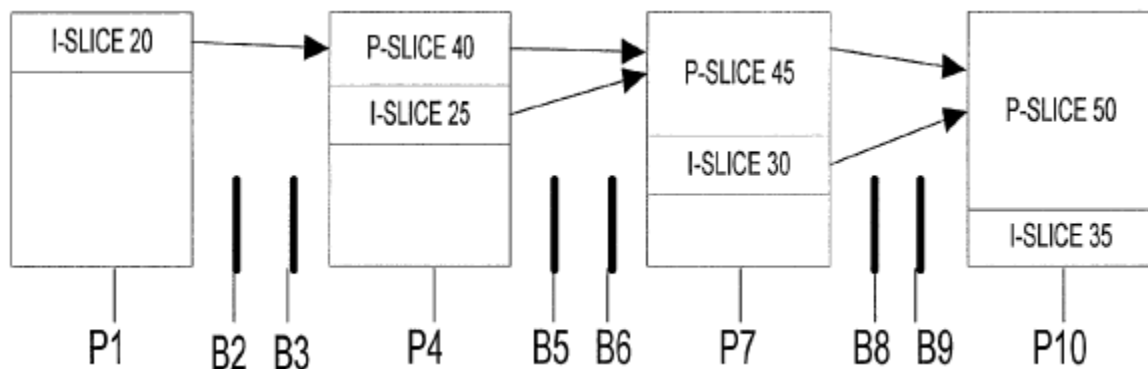
first, second, and third rows in a third picture, etc., *Chen* '677 teaches “decoding the at least one slice in raster order” as required by claim 4. EX-1002 ¶199.

D. Ground 4: *Chen* '677 renders claim 5 obvious.

1. Claim 5

a. *Chen* '677 renders obvious “wherein the picture comprises a P-picture from a HITS stream”

As discussed above, like the '967 patent, *Chen* '677 relates to decoding “progressive I-slice refreshed MPEG data streams.” EX-1004 ¶[0003]; *see also* EX-1001 2:6-16. Fig. 1 shows an example of such a stream, where an I-frame is divided into slices and spread across P-frames:



Chen '677 further explains that “[t]hose skilled in the art will appreciate that the *P*-frames need only be partially decoded in order to recover the I-slices.” *Id.* ¶[0025].

Chen '677 thus teaches that each picture in the stream comprises a P-picture. EX-1002 ¶¶200-01.

While *Chen* '677 does not explicitly state that the “progressive I-slice refreshed MPEG data stream,” in Fig. 1, *see* EX-1004 ¶[0003], is a HITS stream, a POSITA would understand that a HITS stream is but a type of progressive I-slice refreshed MPEG data stream (the type of stream depicted in *Chen* '677). The '967 patent explains as much, stating that HITS streams are “MPEG-2 streams” that “do not include I-pictures” and “use a progressive refresh mechanism.” EX-1001 2:1-16.

Given that the streams in *Chen* '677 are progressive I-slice refreshed MPEG data streams, of which HITS is a type of stream, a POSITA would have found it obvious to implement the teachings of *Chen* '677 over a HITS stream. Such implementation would have been a straightforward substitution. EX-1002 ¶203.

A POSITA thus would have used the teachings of *Chen* '677 to decode HITS streams. *Id.* Implementing *Chen* '677 to operate on HITS streams would have yielded predictable results and known advantages. *Id.*

Accordingly, a POSITA would have found it obvious to implement decoder 220 in *Chen* '677 so as to operate on “a P-picture from a HITS stream,” as required by claim 5. *Id.*

E. Ground 5: The combination of *Chen* '677 and *ISO 13818-2* renders claims 1-5 obvious.

To the extent that Broadcom argues or the Board finds, before or after institution, that *Chen* '677 does not disclose one or more of the claimed “transport packets” (claim 1), or “raster order” (claim 4), *Chen* '677 in view of *ISO 13818-2* renders obvious claims 1-5 with *ISO 13818-2* being relied upon as follows.

1. Claim 1

- a. Element [1.1]: *Chen* '677 in view of *ISO 13818-2* renders obvious “a host processor for transmitting transport packets, said transport packets providing a plurality of instructions.”**

As discussed above, *Chen* '677 discloses a receiver 210 that receives a plurality of instructions in the form of a stream 10 and transmits the instructions to decoder 200. *See, e.g.*, EX-1004 ¶¶[0009], [0025]-[0027], Figs. 2, 3. And because stream 10 *Chen* '677 is an MPEG stream, stream 10 includes the claimed “transport packets.” However, to the extent Broadcom asserts or the Board finds, before or after institution, that *Chen* '677 does not explicitly disclose the claimed “transport packets,” *ISO 13818-2* does.

ISO 13818-2 relates to the ISO/IEC 13818-2 standard, also referred to as the “H.262” and “MPEG-2 Part 2” standard. EX-1009 v (“the identical text is published as ITU-T Rec. *H.262*”), 154 (discussing the complexity of the “MPEG-2

Standard” embodied in the document); *see also* EX-1001 1:30-33 (referring to “the MPEG-2 video compression standard” as “ISO/IEC 13818-2”); EX-1002 ¶207.

ISO 13818-2 discusses sending MPEG streams in the form of one or more “transport packets.” For example, *ISO 13818-2* explains that the data contents of the stream may be sent “within transport packets.” EX-1009 160 (§ D.13.2.2); EX-1002 ¶208. *ISO 13818-2* explains that an encoder that produces a bitstream with “adaptive slice sizes” “keeps track of the data contents within transport packets” and places the start of each slice “at the first opportunity in *every transport packet*.” EX-1009 160 (§ D.13.2.2). *ISO 13818-2* explains that using adaptive slice sizes and placing the start of each slice in every transport packet “gives an advantage in cell or packet based transmission,” such as transmission over the Internet, “or where error detection occurs over a large block of data.” EX-1009 160 (§ D.13.2.2); EX-1002 ¶208.

ISO 13818-2 makes other references to storing and transmitting the stream in packet form. *See, e.g.*, EX-1009 154 (referring to storing information for decoding the bitstream in the “PES packet”), 155; *see also* EX-1011 647, defining “PES” as a “Packetized Elementary Stream” in which a media stream is “formatted into packets.”

A POSITA would have thought to implement stream 10 in *Chen* ‘677 using transport packets. EX-1002 ¶210. Initially, *Chen* ‘677 specifically makes reference

to and builds upon the MPEG-2 standard in *ISO 13818-2*. EX-1004

¶¶[0006]-[0007] (discussing “splicing” of MPEG-2 streams), [0042] (“this algorithm can best be performed in a *processor based MPEG-2 decoder chip*”). Such a modification would thus be a clear design choice that *Chen ‘677* itself suggests. EX-1002 ¶210.

Such a modification would be straightforward to implement. Packetizing data, i.e., reformatting data to place it in the form of packets, was well-known at the time of filing for the ’967 patent, as packet-based transmission enables error correction and retransmission. EX-1011 626 (generally referring to the use of “error correction” and “synchronizing bits” in packets as well as the ability to “identify and correct for lost or errored packets”). And a POSITA would have understood the advantages specifically spelled out in *ISO 13818-2*. EX-1009 160 (§ D.13.2.2); EX-1002 ¶211.

A POSITA would have considered “transport packets providing a plurality of instructions” obvious in view of the combined disclosures of *Chen ‘677* and *ISO 13818-2*. EX-1002 ¶212. Applying the known techniques of *ISO 13838* in *Chen ‘677* would have yielded predictable results and known advantages. EX-1002 ¶212.

Accordingly, a POSITA would have found it obvious to modify stream 10 in *Chen* '677 to include “transport packets, said transport packets providing a plurality of instructions,” as required by element [1.1].

2. Claim 4: *Chen* '677 in view of *ISO 13818-2* renders obvious “wherein decoding the at least one slice above the intracoded slice further comprises causing: decoding the at least one slice in raster order.”

As discussed above, *Chen* '677 teaches slice-by-slice decoding of I-slices stored in top-to-bottom order across P-frames. *See, e.g.*, EX-1004 Fig. 1, ¶¶[0012], [0022]-[0025]. Because decoder 220 in *Chen* '677 decodes the first rows in a first picture, followed by the first and second rows in a second picture, followed by the first, second, and third rows in a third picture, etc., *Chen* '677 teaches “decoding the at least one slice in raster order” as required by claim 4.

However, to the extent that Broadcom argues or the Board finds, before or after institution, that *Chen* '677 does not expressly disclose “decoding the at least one slice in raster order,” *ISO 13818-2* expressly discloses this step.

As explained above (*supra* Ground 2), *ISO 13818-2* dominated distribution of digital video content for TV display at the time the application that matured into the '967 patent was filed, EX-1001 1:31-33. This standard explains that a “slice” is a “series of an arbitrary number of consecutive macroblocks.” EX-1009 18 (§ 6.1.2). Each slice in a frame “shall occur in the bitstream in the order in which

they are encountered, starting at the upper-left of the picture and *proceeding by raster-scan order* from left to right and top to bottom (illustrated in Figures 6-8 and 6-9 as alphabetical order).” *Id*; see again Figure 6-9:

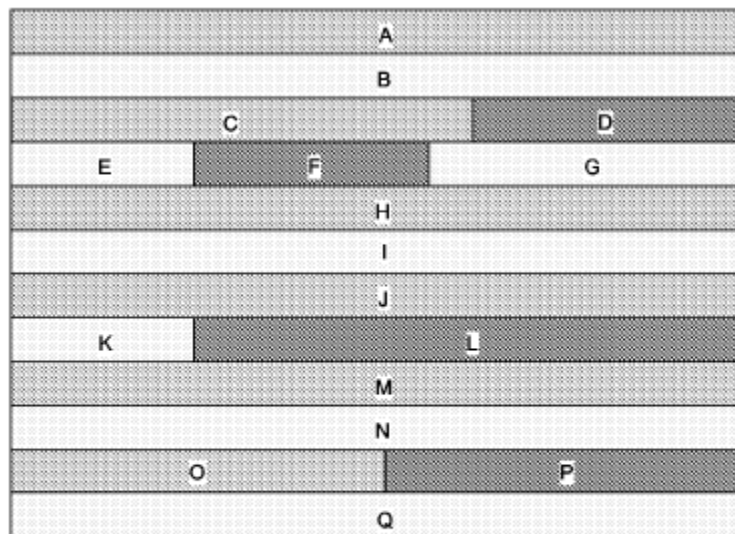


Figure 6-9 – Restricted slice structure

Having started with *ISO 13818-2* when implementing the teachings of *Chen* '677, based on these disclosures of *ISO 13818-2*, a POSITA would have found it obvious to implement the teachings of *Chen* '677 so as to decode at least one slice in raster order. EX-1002 ¶218. Such implementation would have been straightforward. EX-1002 ¶218. Because *ISO 13818-2* explains that MPEG data (such as in *Chen* '677) is encoded in raster order, decoding in raster order would be the most straightforward method of decoding. EX-1002 ¶218. Raster order – e.g., “from top to bottom” – is also a natural method of decoding data because it is a

straightforward application of “first-in, first-out” (FIFO) principles. EX-1002 ¶218; *see also* EX-1010 919 (defining raster order as “the order in which pixels are scanned in a raster display device; usually left-to-right; top-to-bottom”).

A POSITA thus would have considered decoding slices in raster order obvious in view of the combined disclosures of *Chen* '677 and *ISO 13818-2*. Applying the known techniques of *ISO 13838* in *Chen* '677 would have yielded predictable results and known advantages. EX-1002 ¶218. Accordingly, a POSITA would have found it obvious to implement decoder 220 in *Chen* '677 so as to “decod[e] the at least one slice in raster order,” as required by claim 4. EX-1002 ¶218.

VI. MANDATORY NOTICES

A. Real Party-in-Interest

The real parties in interest are LG Electronics, Inc., and LG Electronics U.S.A., Inc.

B. Related Matters

The '967 patent is involved in the following litigations:

Name	Number	District	Filed
<i>Broadcom Corporation v. VIZIO, Inc.</i>	8-17-cv-00408	CACD	March 7, 2017
<i>Broadcom Corporation v. LG Electronics Inc. et. al.</i>	8-17-cv-00404	CACD	March 7, 2017
<i>Broadcom Corporation v. MediaTek Inc., et al.</i>	8-17-cv-00405	CACD	March 7, 2017
<i>Certain Semiconductor Devices and Consumer Audiovisual Products Containing the Same</i>	ITC-337-TA-1047	ITC	March 7, 2017

C. Lead and Back-Up Counsel, and Service Information

Lead Counsel: Erika H. Arner (Reg. No. 57,540; Tel. (571) 203-2754; erika.arner@finnegan.com), attorney at Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P., 11955 Freedom Drive, Reston, VA 20190-5675.

Backup Counsel: Joshua L. Goldberg (Reg. No. 59,369; Tel. (202) 408-6092; joshua.goldberg@finnegan.com), attorney at Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P., 901 New York Avenue, N.W., Washington, DC 20001-4413.

Backup Counsel: Christopher C. Johns (Reg. No. 68,664; Tel. (202) 408-4188; christopher.johns@finnegan.com), attorney at Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P., 901 New York Avenue, N.W., Washington, DC 20001-4413.

LGE consents to e-mail service at these e-mail addresses and LG-Broadcom-IPR@finnegan.com.

VII. CERTIFICATION UNDER 37 C.F.R. § 42.24(d)

This petition complies with the requirements of 37 C.F.R. § 42.24. As calculated by the word count feature of Microsoft Word 2016, it contains 13,857 words, excluding the words contained in the following: Table of Contents, Table of Authorities, List of Exhibits, Mandatory Notices, Certification Under §42.24(d), and Certificate of Service.

VIII. GROUNDS FOR STANDING

Petitioner certifies the '967 patent is available for *inter partes* review and that Petitioner is not barred or estopped from requesting *inter partes* review of the '967 patent challenging the patent claims on the grounds identified in this petition.

IX. STATEMENT OF PRECISE RELIEF REQUESTED FOR EACH CLAIM CHALLENGED

LGE requests review of claims 1-5 based on each of the above grounds. Claims 1-5 are unpatentable under 35 U.S.C. §§ 102 and 103. The claim

construction, reasons for unpatentability, and specific evidence supporting this request are detailed above.

X. CONCLUSION

For the reasons set forth above, the challenged claims 1-5 are unpatentable, so trial should be instituted and the claims should be cancelled.

Respectfully submitted,

Dated: June 13, 2017

By: /Erika H. Arner/
Erika H. Arner, Lead Counsel
Reg. No. 57,540

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. §§ 42.6(e) and 42.105(b), the undersigned certifies that the foregoing **Petition for *Inter Partes* Review, the associated LGE Exhibits 1001-1014, and Petitioner's Power of Attorney** were served on June 13, 2017, by FedEx at the following address of record for the subject patent.

Scott A. Horstemeyer
Thomas Horstemeyer, LLP (Broadcom)
400 Interstate North Parkway, SE
Suite 1500
Atlanta GA 30339
(770)933-9500

Dated: June 13, 2017

By: /Lauren K. Young/
Lauren K. Young
Litigation Legal Assistant
FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.