

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE PATENT TRIAL AND APPEAL BOARD

---

LG ELECTRONICS, INC.,  
Petitioner,

v.

BROADCOM CORPORATION,  
Patent Owner.

---

Case IPR2017-01544  
Patent 7,342,967

---

**PATENT OWNER'S RESPONSE**

**TABLE OF CONTENTS**

	<b>Page</b>
<b>I. THE INVENTIONS OF THE '967 PATENT .....</b>	<b>2</b>
<b>II. THE ASSERTED PRIOR ART .....</b>	<b>4</b>
<b>III. CLAIM CONSTRUCTION.....</b>	<b>5</b>
<b>A. “host processor” .....</b>	<b>6</b>
<b>B. “instructions” .....</b>	<b>8</b>
<b>IV. THE CHEN-BASED GROUNDS (3-5) DO NOT RENDER THE CHALLENGED CLAIMS UNPATENTABLE.....</b>	<b>10</b>
<b>A. Chen does not anticipate claim 1 .....</b>	<b>10</b>
<b>B. Chen does not anticipate claims 2-5 .....</b>	<b>16</b>
<b>C. Chen alone and the combination of Chen and ISO 13818-2 do not render obvious claims 1-5 .....</b>	<b>16</b>
<b>V. THE MACINNIS-BASED GROUNDS (1-2) DO NOT RENDER THE CHALLENGED CLAIMS UNPATENTABLE .....</b>	<b>16</b>
<b>A. MacInnis does not anticipate claim 1 .....</b>	<b>19</b>
<b>B. The combination of MacInnis and ISO 13818-2 does not render obvious claim 4 .....</b>	<b>24</b>
<b>VI. CONCLUSION .....</b>	<b>25</b>

**TABLE OF AUTHORITIES**

	<b>Page(s)</b>
<b>Cases</b>	
<i>In re Cuozzo Speed Techs., LLC</i> , 793 F.3d 1268 (Fed. Cir. 2015) .....	6
<i>Cuozzo Speed Techs., LLC v. Lee</i> , 136 S. Ct. 2131 (2016).....	6
<i>Dynamic Drinkware, LLC v. Nat’l Graphics, Inc.</i> , 800 F.3d 1375 (Fed. Cir. 2015) .....	5
<i>Microsoft Corp. v. Proxyconn, Inc.</i> , 789 F. 3d 1292 (Fed. Cir. 2015) .....	6
<i>Net MoneyIN, Inc. v. VeriSign, Inc.</i> , 545 F.3d 1359 (Fed. Cir. 2008) .....	22
<i>Nichia Corp. v. Emcore Corp.</i> , Case IPR2012-00005, 2014 WL 574596 (PTAB, Feb. 11, 2014).....	6
<i>In re NTP, Inc.</i> , 654 F.3d 1279 (Fed. Cir. 2011) .....	6
<i>PPC Broadband, Inc. v. Corning Optical Communications RF</i> , 815 F. 3d 747 (Fed. Cir. 2016) .....	6
<i>In re Skvorecz</i> , 580 F.3d 1262 (Fed. Cir. 2009) .....	6
<b>Statutes</b>	
35 U.S.C. § 102 .....	22
35 U.S.C. § 316(e) .....	4

Patent Owner Broadcom Corporation (“Broadcom”) respectfully submits this Response to Petitioner LG Electronics, Inc.’s (“Petitioner” or “LG”) Petition for *Inter Partes* Review (“Petition”) of Broadcom’s U.S. Patent No. 7,342,967 (the “’967 Patent” or Ex. 1001).

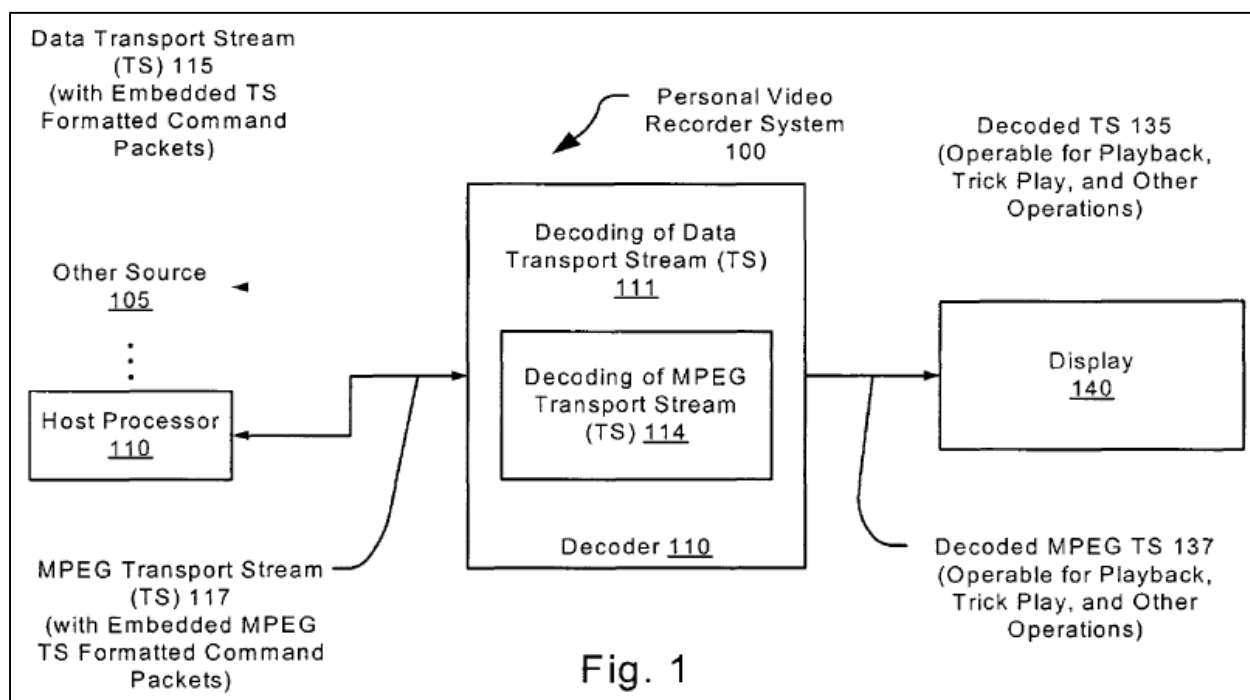
The ’967 Patent discloses and claims a novel and innovative system for decoding and displaying pictures that are transmitted as compressed data in transport packets. Specifically, the ’967 Patent’s system is directed to a system that processes and displays a special type of encoded data stream known as a Headend In The Sky (HITS) stream. Rather than having complete reference pictures, known as I-pictures in the MPEG-2 standard, a HITS stream uses a series of I-slices that are spread out through a series of consecutive pictures. Central to the ’967 Patent’s system is instructions in the transport packets that are executed by the system’s video decoder to perform a specific decoding procedure for HITS streams. The decoding procedure involves decoding slices above an I-slice and the I-slice but without decoding the slices below the I-slice. This procedure “advantageously reduces the processing required to build a clean reference picture.”

All five of LG’s grounds of unpatentability fail to show by a preponderance of the evidence that claims 1-5 of the ’967 Patent are unpatentable. LG’s unpatentability arguments are based on two primary references (U.S. Patent

Application Publication No. 2001/0026677 – “Chen ’677” and U.S. Patent Application Publication No. 2002/0061183 – “MacInnis”) and a secondary reference (ISO/IEC Standard 13818-2:2000 – “ISO 13818-2”). As discussed in detail below, LG’s arguments fail to prove unpatentability because they do not explain how (1) Chen ’677 discloses “instructions” that are executed to cause the claimed video decoding process; (2) Chen ’677 discloses a “host processor”; or (3) MacInnis discloses “transport packets providing a plurality of instructions ... wherein execution of the instructions by the video decoder comprises causing ....” Based on LG’s failure to show that independent claim 1 is unpatentable, the Board should find that claims 1-5 are not unpatentable.

## **I. THE INVENTIONS OF THE ’967 PATENT**

The ’967 patent generally relates to a system for displaying pictures that include, among other components, a host processor and a video decoder. As shown in Figure 1 below, the host processor transmits a data transport stream to a decoder and controls the playback (including trick playback) of the data. Ex. 1001, 3:45–50. The data transport stream is packetized, such that the stream is a series of transport packets. *Id.* at 4:23-24. The data transport stream also includes instructions for performing tasks such as trick play functionality. *Id.* at 5:7-13. Based on the instructions in the transport stream, the decoder then decodes the data transport stream. *Id.* at 3:53-57.



*Id.* at Figure 1.

In the '967 Patent's system, the instructions in the transport packets command the system's video decoder to perform a specific decoding procedure for HITS streams. *Id.* at 5:7-13. The decoding procedure involves decoding slices above an I-slice and the I-slice but without decoding the slices below the I-slice. *Id.* at 5:37-7:12. The execution of the instructions causes the decoder to (1) select a picture comprising an intracoded slices, at least a slice above the intracoded slice, and at least a slice below the intracoded slice; (2) decode the intracoded slice and the slice above the intracoded slice; and (3) decode at least a portion of another picture after decoding the intracoded slice and the slice above the intracoded slices, without having decoded the slice below the intracoded slice. *Id.* at 5:7-13. The

patent explains that by not “decoding the portion of the P-pictures below the intracoded slice advantageously reduces the processing required to build a clean reference picture.” *Id.* at 3:8-10.

Claim 1 of the '967 patent is the only independent claim and is reproduced below:

1. A system for displaying pictures, said system comprising:
  - a host processor for transmitting transport packets, said transport packets providing a plurality of instructions;
  - a video decoder for executing the plurality of instructions; and
  - wherein execution of the instructions by the video decoder comprises causing:
    - 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;
    - decoding the at least one slice above the intracoded slice;
    - decoding the intracoded slice; and
    - 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.

## **II. THE ASSERTED PRIOR ART**

“In an *inter partes* review, the burden of persuasion is on the petitioner to prove ‘unpatentability by a preponderance of the evidence,’ 35 U.S.C. § 316(e),

and that burden never shifts to the patentee.” *Dynamic Drinkware, LLC v. Nat'l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015). LG’s unpatentability arguments are based on two primary references (Chen ’677 and MacInnis) and a secondary reference (ISO 13818-2).<sup>1</sup>

Ground 1 asserts that MacInnis anticipates claims 1-5.

Ground 2 asserts that the combination of MacInnis and ISO 13818-2 renders obvious claim 4.

Ground 3 asserts that Chen ’677 anticipates claims 1-4.

Ground 4 asserts that Chen ’677 renders obvious claim 5.

Ground 5 asserts that the combination of Chen ’677 and ISO 13818-2 renders obvious claims 1-5.

### **III. THE LEVEL OF ORDINARY SKILL IN THE ART**

A POSA at the time of the invention of the ’967 patent would have had a Bachelor’s Degree in Electrical Engineering, Computer Science, or a similar discipline, with one to two years of experience in this or a related field. The POSA would also have been familiar with software or hardware related to digital signal, image, and video processing. Ex. 2002, ¶ 21.

### **IV. CLAIM CONSTRUCTION**

In *inter partes* review, claims in an unexpired patent are given their

---

<sup>1</sup> MacInnis is Ex. 1003, Chen ’677 is Ex. 1004, and ISO 13818-2 is Ex. 1009.



“broadest reasonable construction in light of the specification of the patent in which it appears.” *Cuozzo Speed Techs., LLC v. Lee*, 136 S. Ct. 2131, 2133 (2016) (citing 37 C.F.R. § 42.100(b)). This is the broadest reasonable construction that a person of ordinary skill in the art would give the term in the context of the specification and its ordinary meaning in the art. *See, e.g., Nichia Corp. v. Emcore Corp.*, Case IPR2012-00005, 2014 WL 574596, at \*4 (PTAB, Feb. 11, 2014); *In re Cuozzo Speed Techs., LLC*, 793 F.3d 1268, 1275 (Fed. Cir. 2015).

“[The] construction cannot be divorced from the specification and the record evidence.” *In re NTP, Inc.*, 654 F.3d 1279, 1288 (Fed. Cir. 2011). “Above all, the broadest reasonable interpretation must be reasonable in light of the claims and specification.” *PPC Broadband, Inc. v. Corning Optical Communications RF*, 815 F. 3d 747, 755 (Fed. Cir. 2016). “A construction that is “unreasonably broad” and which does not “reasonably reflect the plain language and disclosure” will not pass muster.” *Microsoft Corp. v. Proxyconn, Inc.*, 789 F. 3d 1292, 1298 (Fed. Cir. 2015). Similarly, “giving claims their broadest reasonable interpretation . . . does not include giving claims a legally incorrect interpretation.” *In re Skvorecz*, 580 F.3d 1262, 1267 (Fed. Cir. 2009).

**A. “host processor”**

The '967 Patent uses the term “host processor” in accordance with its plain and ordinary meaning and does not impose any special definition on the term. Dr.

Richardson explains that a person of ordinary skill in the art would have understood that a “host processor” is “a central processor responsible for the overall control of the system.” Ex. 2002, ¶ 24 citing Ex. 2003, p. 5 (defining a “host processor” as “A processor that controls all or part of a user application network.”). Dr. Delp (LG’s expert) admitted that the term “host processor” was a common term to a POSA outside the context of the ’967 Patent. Ex. 2005, 19:9-15. Dr. Richardson’s understanding of the plain and ordinary meaning of the term “host processor” is consistent with its use in the ’967 patent. When discussing Figure 1, the ’967 patent specification states that “The host processor 110 or the any other source 105 is the device **controlling** the playback (including trick play playback) of the data.” Ex. 1001, 3:47-50 (emphasis added). The host processors in other figures (e.g., Figures 2 and 3) further demonstrate that a host processor (CPU 240 in Figure 2 and CPU 390 in Figure 3) has overall control of the system by controlling other components throughout the system. *Id.* at Figures 2 and 3. The ’967 Patent also explains that a host processor can control the system by inserting command packets into the transport stream. *Id.* at 7:37-43. Accordingly, the term “host processor” should be given its plain and ordinary meaning, which is “a central processor responsible for the overall control of the system.”

In its institution decision, the Board construed the term “host processor” to mean “a device or element that delivers an incoming transport stream (i.e., a

stream of packets) that may contain instructions for controlling the playback to a decoder device (such as any audio or visual decoding device or functionality).” Institution Decision at 10. However, this construction fails to account for the essential control aspect that makes a generic electronic component a “host processor.” The Board’s construction fails to take into account that a processor must *processes* data and that the processor must be a “host” processor, meaning it exerts control over a system. Ex. 2002, ¶ 25. When generally discussing a “host processor,” Dr. Delp described a “processor” as “a device that processes, handles data.” Ex. 2005, 20:3-15. Dr. Delp also described a “host” processor as “the main processor.” *Id.* at 20:16-21:1. Prior to providing its proposed construction, the Board explicitly noted that the “host processor” must perform the function of “*controlling* the playback (including trick playback) of the data.” *Id.* (emphasis added). However, the Board’s proposed construction ignores the required control functionality that even it noted was necessary for a component to be a “host process.” Accordingly, Dr. Richardson’s proposed plain and ordinary meaning should be adopted.

**B. “instructions”**

The ’967 Patent uses the term “instructions” in accordance with its plain and ordinary meaning and does not impose any special definition on the term. In the context of the ’967 Patent, Dr. Richardson explains that a person of ordinary skill

in the art would have understood that “instructions” are “information or syntax that affects the operation of a video decoder.” Ex. 2002, ¶ 26 citing Ex. 2004, p. 3 (defining “instructions” as “A pattern of digits which signifies to a computer that a *particular* operation is to be performed ....”) (emphasis added). Dr. Richardson’s understanding of the plain and ordinary meaning of the term “instructions” is consistent with its use in the ’967 patent. Ex. 1001, 5:12-13, 7:32-36, claim 1.

The plain and ordinary meaning proposed by Dr. Richardson is consistent with the Board’s construction for the term as provided in its institution decision – “commands that specify playback functions to be performed, including trick play functionality.” However, Dr. Richardson’s proposed construction does not limit the term to “commands” and does not limit the instructions to “specify[ing] playback functions.” The use of the word “command” versus more general words such as “information or syntax” can imply that the instructions must be command packets, which is not supported by the specification because the instructions are included in a command packet. *Id.* at 5:12-13. Also, the use of the words “that specify playback functions” versus more general words such as “that affects the operation” unnecessarily limits the content of the instructions to specifying the playback function, whereas the ’967 Patent specification states that the instructions simply alter the operation of the video decoder to perform different operations. *Id.* at 5:12-1, 6:55-7:12, claim 1. Accordingly, Dr. Richardson’s proposed plain and

ordinary meaning should be adopted.

**V. THE CHEN-BASED GROUNDS (3-5) DO NOT RENDER THE CHALLENGED CLAIMS UNPATENTABLE**

**A. Chen does not anticipate claim 1**

Chen '677's alleged invention is a system that receives "progressive I-slice refreshed MPEG data streams" and transcodes the streams to "I-frame based MPEG data streams." Ex. 1004, ¶¶ [0012], [0027]. Chen '677 explains the transcoding process as follows:

A progressive I-slice refreshed MPEG data stream having I -slices distributed over multiple P-frames is received by a television appliance. The P-frames are decoded by the decoder to recover the I-slices which make up a complete I-frame. The recovered I-slices are assembled (e.g., by processor) into a complete I-frame. The complete I-frame is encoded at an encoder. A selected P-frame in the MPEG data stream is replaced with the encoded I-frame to provide an encoded I-frame based data stream. The I-frame based data stream is stored for trick play mode use.

*Id.* at ¶ [0012]. Accordingly, Chen '677's system simply reformats streams of one type (i.e., "progressive I-slice refreshed MPEG data stream") to another type (i.e., "I-frame based MPEG data streams"). Ex. 2002, ¶ 28.

Chen '677 further explains that "The I-frame based data stream is stored for trick play mode use." *Id.* at ¶¶ [0012],[0027]. Thus, it is the "I-frame based MPEG data streams," not the "progressive I-slice refreshed MPEG data streams,"

that are decoded and played during trick play operations.

**1. Chen '677 does not disclose “instructions” that are executed to cause the claimed video decoding process**

LG’s argument that Chen '677 discloses “instructions” that are executed to cause the claimed video decoding process fails because (1) LG assume that the video stream causes the decoder to decode in a certain way that is dependent on the content of the video stream, which is not supported by Chen '677, and (2) Chen '677’s “progressive I-slice refreshed MPEG data stream” does not include “instructions” under either Broadcom’s or the Board’s construction of that term.

In its Petition, LG argues that: “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” ....” Petition at 56. However, LG points to no evidence that “the stream causes decoder 220 to decode it” as LG alleges. *Id.* There are a variety of methods that the Chen '677 system could use to cause the video decoder to start decoding and/or to carry out decoding in a certain way. Ex. 2002, ¶¶ 36-38. None of the many possible methods for starting the video decoder or carrying out decoding in different ways are discussed in Chen '677, in LG’s Petition, or Dr. Delp’s expert declaration. Accordingly, LG has not identified instructions that cause the claimed video decoding process.

Even if “the stream causes decoder 220 to decode it” as LG alleges, the stream does not contain “instructions” as construed by Broadcom or the Board. The Chen ’677 system identified in LG’s petition performs the same transcoding process on every progressive I-slice refreshed MPEG data stream that enters the system and, therefore, does not disclose “instructions” under Broadcom’s construction. Broadcom’s proposed construction states that instructions are “information or syntax that *affects* the operation of a video decoder.” LG has not identified anything in the video stream that affects the operation of the video decoder. Petition at 55-57. Chen ’677 explains that “The P-frames are decoded at decoder 220 to recover the I-slices which make up a complete I-frame.” Ex. 1004, ¶ [0027]. Nothing in the progressive I-slice refreshed MPEG data stream affects this operation or causes it to change in any way dependent on the content of the progressive I-slice refreshed MPEG data stream. The Chen ’677 system operates in an identical manner regardless of the playback mode. Ex. 2002, ¶ 39. That is, the Chen ’677 video decoder operates the same regardless of whether the stream is going to be store for future trick mode operations or whether the stream is going to be played back in normal speed. *Id.* The Chen ’677 system simply “*enable[s]* trick play modes on a television appliance,” it does not perform different operations for trick mode operations. Ex. 1004, ¶ [0012]. Accordingly, LG has not identified any “instructions” that cause the claimed video decoding process.

Additionally, the Chen '677 system identified in LG's petition does not disclose "instructions" under the Board's construction because the incoming progressive I-slice refreshed MPEG data stream does not contain commands that specify playback functions to be performed. In its institution decision, the Board stated that "we determine on this record that Chen '677 meets this limitation, for Chen '677's data stream contains commands that specify trick play functions to be performed." Institution Decision at 16 citing Ex. 1004 ¶ [0027]. However, in Chen '677's system the incoming "progressive I-slice refreshed MPEG data streams" do not include any commands that specify playback functions, or indeed any commands at all. Ex. 1004, ¶¶ [0025]-[0027]. As discussed above, Chen '677's system simply "*enable[s]* trick play modes on a television appliance" by transcoding "progressive I-slice refreshed MPEG data streams" to "I-frame based MPEG data streams." *Id.* at ¶ [0012]. Chen '677 provides no discussion of "commands that specify playback functions." Chen '677 does not describe how the video is played back on a display, let alone how trick play playback operations are implemented using the "I-frame based MPEG data streams." Accordingly, LG has not identified any "instructions" that cause the claimed video decoding process.

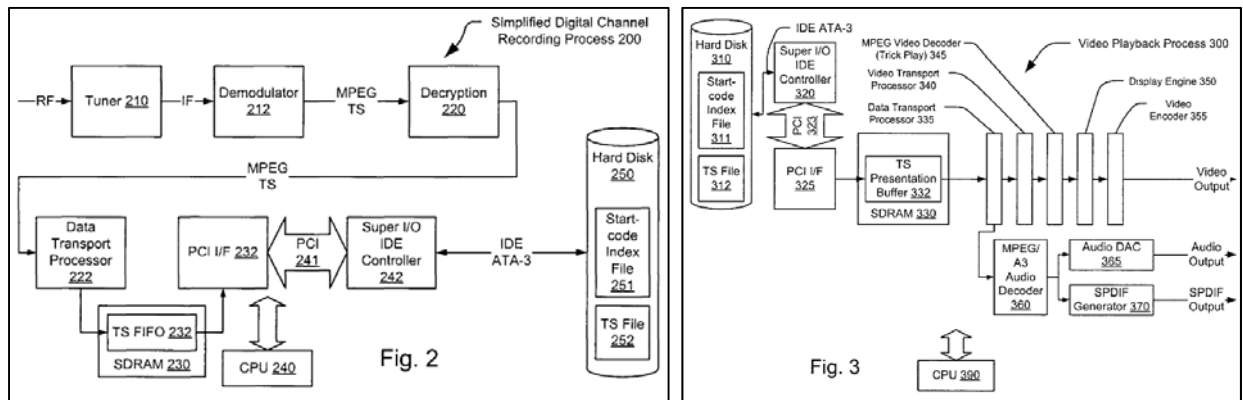
Accordingly, claim 1 is not unpatentable as being anticipated by Chen '677 because Chen '677 fails to disclose "instructions" that are executed to cause the



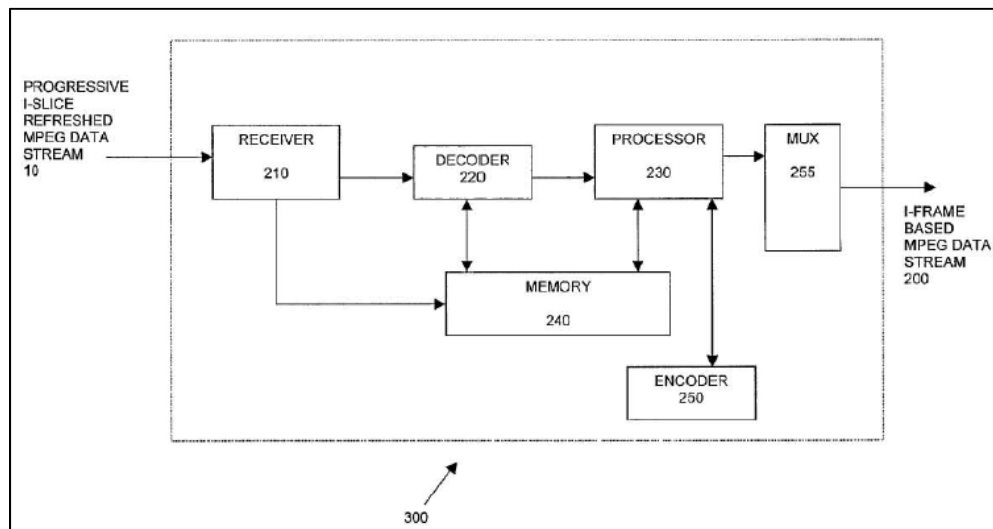
claimed video decoding process.

## 2. Chen '677 does not disclose a "host processor"

Chen '677 does not disclose a "host processor" under Broadcom's construction because the receiver 210 is not "a central processor responsible for the overall control of the system." As shown in Figures 2 and 3 below, the '967 Patent's system includes a central processor responsible for the overall control of the system, such as CPU 240 and processor/CPU 390.



Ex. 1001, Figures 2, 3. Whereas Chen '677's system does not include a central processor responsible for the overall control of the system, as shown in Figure 3 below.



Ex. 1004, Figure 3. Chen '677 **does not describe** the receiver 210 as a “processor” but **does describe** another component, “processor 230,” as a “processor.” *Id.* at ¶ [0027], Figure 3. This shows that Chen '677 does not even consider the receiver 210 a “processor” based on a plain and ordinary meaning of that word. As Figure 3 of Chen '677 shows, the receiver 210 only moves the progressive I-slice refreshed MPEG data stream data to the decoder 220 and/or memory 240. *Id.* at Figure 3. The only discussion in Chen '677 of the receiver 210's responsibilities states that receiver 210 receives the progressive I-slice refreshed MPEG data stream. *Id.* at ¶ [0027] (“A 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).”) Chen '677's does not expressly describe the receiver 210 as performing any processing functions. Ex. 2002, ¶ 42. Even if it is assumed that the receiver 210 is a “processor,” it is not a “host” processor

because the receiver 210 does not provide any control over the system.

Accordingly, LG has not identified a “host processor” in Chen ’677.

Accordingly, claim 1 is not unpatentable as being anticipated by Chen ’677 because Chen ’677 fails to disclose a host processor.

**B. Chen does not anticipate claims 2-5**

As discussed above, claim 1 is not anticipated by Chen. Claims 2-5 depend from claim 1 and, therefore, are not unpatentable as being anticipated by Chen because claim 1 is not anticipated by Chen.

**C. Chen alone and the combination of Chen and ISO 13818-2 do not render obvious claims 1-5**

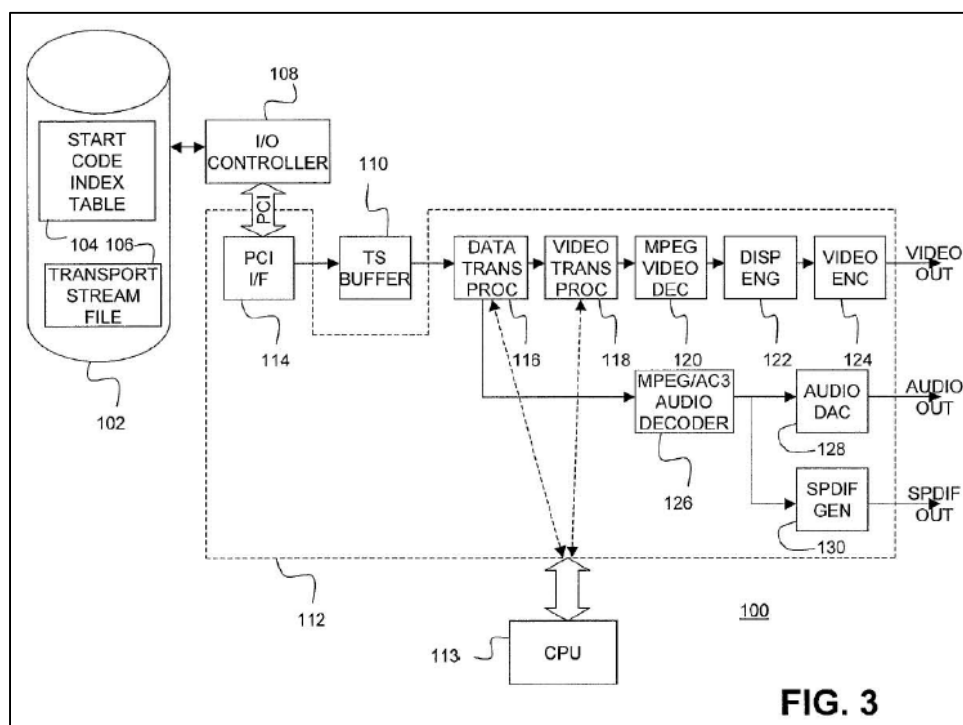
LG’s obviousness arguments presented as Ground 4 and Ground 5 do not address the deficiencies in Chen ’677 that are discussed above. Nothing in LG’s proposed combination of Chen ’677 and ISO 13818-2 addresses the deficiencies discussed above. Accordingly, based on LG’s failure to explain how Chen ’677 discloses either “instructions” that are executed to cause the claimed video decoding process or a “host processor,” the Board should find that Chen ’677 alone and the combination of Chen ’677 and ISO 13818-2 do not render unpatentable claim 1-5 of the ’967 Patent.

**VI. THE MACINNIS-BASED GROUNDS (1-2) DO NOT RENDER THE CHALLENGED CLAIMS UNPATENTABLE**

MacInnis explains that his “invention relates generally to the decoding and

presentation of digital audio-video streams, and more particularly to a system and method for implementing ‘trick modes’ of personal video recording (PVR), such as, for example, fast forward play and reverse play at various speeds for digital audio-video streams.” Ex. 1003, ¶ [0002].

MacInnis explains that in one embodiment, the CPU 113 implements the trick modes by controlling the data transport processor 116 and/or the video transport processor 118. *Id.* at ¶ [0037]. MacInnis further explains that “a data transport driver (DID) which preferably runs in the CPU 113, preferably 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.* at ¶ [0040]. MacInnis explains that the CPU can control the video transport processor 118 with command transport packets that are inserted into the transport stream. *Id.* The CPU’s control over the data transport processor 116 and the video transport processor 118 is illustrated in Figure 3 below:



*Id.* at Figure 3. MacInnis also states that the CPU’s DTD can use the command packets to send “various commands” to the video decoder 120, but does not expressly state what those commands are or how they affect the system. *Id.* at ¶ [0040].

MacInnis states that “[i]n other embodiments, various different software, hardware and/or firmware available to those skilled in the art may be used to provide commands to the data transport processor 116, the video transport processor 118 and/or the video decoder 120 to implement and control the trick modes.” *Id.* at ¶ [0040]. Throughout the rest of the specification, MacInnis then describe how the CPU 113 controls the video decoder 120 through software-based interactions known as application program interface (API) calls, not command

packets. *Id.* at ¶¶ [0101], [0127], [0134], [0154], [0155], [0157], [0165].

**A. MacInnis does not anticipate claim 1**

LG's Petition fails to establish unpatentability of claim 1 by a preponderance of the evidence because it does not explain how MacInnis expressly or inherently discloses "transport packets providing a plurality of instructions ... wherein execution of the instructions by the video decoder comprises causing ..." the video decoding process claimed in claim 1. Notably, claim 1 requires a video decoder that executes "instructions" that are provided by transport packets. As discussed in detail below, the MacInnis citations relied on by LG and the Board in its institution decision do not expressly or inherently disclose (1) that transport packets with instructions are executed by the video decoder or (2) that the instructions cause the video decoder to perform the claimed decoding process. As further discussed below, to the extent MacInnis describes any communications between the host processor and the video decoder that control the video decoder's operations, such communications are described as being carried out by software-based communications known as application program interface (API) calls, not by command packets in a transport stream.

**1. MacInnis does not expressly or inherently disclose that the video decoder executes “command transport packets” that cause the video decoder to perform the claimed decoding process**

LG’s citations to MacInnis’s discussion of “command transport packets” do not expressly or inherently disclose “transport packets providing a plurality of instructions ... wherein execution of the instructions by the video decoder comprises causing ...” the video decoding process claimed in claim 1. Petition at 25-29. LG primary relies on ¶ [0040], which states that the CPU’s DTD “inserts a command transport packet containing various commands to the video transport processor 118 and the video decoder 120.” Ex. 1003, ¶ [0040]. MacInnis does not expressly state what the “commands” are except to state that “[t]he inserted transport packet preferably contains command fields for controlling the data flow of the MPEG video stream during play back ....” *Id.* However, MacInnis explains that commands for controlling “the data flow” are sent to the data transport processor 116 and/or the video transport processor 118 only and not to the video decoder 120. *Id.* (“data transport driver (DTD) which preferably runs in the CPU 113, preferably 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.* at ¶ [0037] (“The 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).”) Accordingly, MacInnis does not expressly state what, if any, commands are sent to the video decoder 120.

LG then assumes without any supporting evidence in MacInnis that these “various commands” control the video decoder 120 and cause the video decoder to perform the decoding process that is claimed in claim 1. Petition at 29. MacInnis describes the decoder’s trick modes operations in Section III “Normal/Slow/Fast Forward” (§ [0072]) and Section IV “Reverse” (§ [0105]). And nowhere in these sections does MacInnis state that any video decoder operations are caused by the video decoder executing instructions from a command packet. *See* Ex. 1003, §§ [0072]-[0134]. In fact, as discussed in more detail below, MacInnis refers to API calls as controlling the video decoder’s operations, not command packets.

LG also argues, under their heading “Construction II,” that the video stream itself is the claimed “instructions.” Petition at 27-28. However, LG never argues that these “instructions” are what cause the video decoder to perform the claimed decoding process. Petition at 29. Instead, LG only argues that the “command fields in transport packets” are the alleged “instructions” that cause the video decoder to perform the claimed decoding process. Accordingly, whether the video stream itself is the alleged “instructions” is a moot point because LG does not argue or explain how these instructions cause the video decoder to perform the claimed decoding process.



Furthermore, LG's citations to ¶ [0040] in combination with ¶ [0043] are improper because MacInnis explicitly refers to the embodiment in ¶ [0040] as distinct from the embodiments in ¶ [0043]. Specifically, MacInnis states that ¶ [0040] is "an embodiment according to the present invention" and directly after discussing this embodiment states that ¶ [0043] is related to "other embodiments." It is improper to pick and choose teachings from different embodiments, even if from the same reference, to establish anticipation. "[T]he hallmark of anticipation is prior invention, the prior art reference--in order to anticipate under 35 U.S.C. § 102--must not only disclose all elements of the claim within the four corners of the document, but must also disclose those elements arranged as in the claim." *Net MoneyIN, Inc. v. VeriSign, Inc.*, 545 F.3d 1359, 1369 (Fed. Cir. 2008) (internal quotations removed). Even if ¶ [0043] were to be considered the same embodiment as the embodiment as described in ¶ [0040], ¶ [0043] only vaguely refers to "command transport packets" and does not expressly or inherently disclose "transport packets providing a plurality of instructions ... wherein execution of the instructions by the video decoder comprises causing ..." the video decoding process claimed in claim 1.

Accordingly, claim 1 is not unpatentable as being anticipated by MacInnis because MacInnis does not disclose "transport packets providing a plurality of instructions ... wherein execution of the instructions by the video decoder

comprises causing ...” the video decoding process claimed in claim 1.

**2. MacInnis explains that the host processor controls the video decoder through API calls and not instructions in transport packets**

Rather than executing instructions via command packets, MacInnis explains that “The preferred means to control the decoder is for the host to tell the decoder, via an API call, to start decoding ....” Ex. 1004, ¶ [0154]. Directly after MacInnis describes the specific embodiment that includes “command transport packets,” MacInnis explains that “In *other embodiments*, various different software, hardware and/or firmware available to those skilled in the art may be used to provide commands to the data transport processor 116, the video transport processor 118 and/or the video decoder 120 to implement and control the trick modes.” Ex. 1003, ¶ [0043] (emphasis added). MacInnis then repeatedly describes the host as communicating and controlling the video decoder via API calls, not command packets. *Id.* at ¶¶ [0101], [0127], [0134], [0154], [0155], [0157], [0165].

Specifically, MacInnis explains that both fast-forward and reverse trick mode operations in the video decoder are caused by API calls from the host. LG points to ¶ [0095] and ¶ [0111] of MacInnis as describing the claimed decoding process where the decoder performs “decoding at least a portion of another picture after decoding the at least one slice above the intracoded slice and the intracode

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

Petition at 37-38. These paragraphs in MacInnis describe fast-forward (§ [0095]) and reverse (§ [0111]) operations in the video decoder. Ex. 1003, § [0089] (the header for the section containing § [0095]) and § [0108] (the header for the section containing § [0111]). However, MacInnis explains that the host controls the video decoder’s fast forward operations through API calls and not instructions in command packets: “In this case of fast forward with skipping... the host preferably tells the decoder, via an API call ....” *Id.* at § [0157]. Similarly, MacInnis explains that the host controls the video decoder’s reverse operations through API calls and not instructions in command packets: “The host may command the decoder to do this [reverse play] via an API call.” *Id.* at § [0165].

Accordingly, claim 1 is not unpatentable as being anticipated by MacInnis because MacInnis discloses that API calls from the host processor to the video decoder control the video decoder’s operations and, therefore, MacInnis does not disclose “transport packets providing a plurality of instructions ... wherein execution of the instructions by the video decoder comprises causing ...” the video decoding process claimed in claim 1.

**B. The combination of MacInnis and ISO 13818-2 does not render obvious claim 4**

LG’s obviousness argument presented as Ground 2 does not address the

deficiencies in MacInnis that were discussed above and nothing in the proposed combination of MacInnis and ISO 13818-2 addresses these deficiencies.

Accordingly, based on LG's failure to explain how MacInnis discloses "wherein execution of the instructions by the video decoder comprises causing," the Board should find that the combination of MacInnis and ISO 13818-2 does not render unpatentable claim 4 of the '967 Patent.

## **VII. CONCLUSION**

For the reasons discussed above, LG failed to demonstrate that claims 1-5 are unpatentable by a preponderance of the evidence.

Respectfully submitted,

Dated: March 19, 2018

/John M. Caracappa/  
John M. Caracappa  
Reg. No. 43,532  
Thomas C. Yebernetsky  
Reg. No. 70,418  
STEPTOE & JOHNSON LLP  
1330 Connecticut Ave., N.W.  
Washington, D.C. 20036  
[jcaracp@steptoe.com](mailto:jcaracp@steptoe.com)  
[tyeburnetsky@steptoe.com](mailto:tyeburnetsky@steptoe.com)

Attorney for Patent Owner  
Broadcom Corporation

**CERTIFICATE OF COMPLIANCE**

The undersigned hereby certifies that the foregoing PATENT OWNER'S  
RESPONSE contains 5,141 words, excluding those portions identified in 37 C.F.R.  
§ 42.24(a), as measured by the word-processing system used to prepare this paper.

Dated: March 19, 2018

/John M. Caracappa/  
John M. Caracappa  
Reg. No. 43,532

Attorney for Patent Owner  
Broadcom Corporation

**CERTIFICATE OF SERVICE**

Pursuant to 37 C.F.R. § 42.6(e), the undersigned certifies service of  
PATENT OWNER'S RESPONSE on the counsel of record for the Petitioner by  
filing this document through PTAB E2E system on March 19, 2018 and served a  
true and correct copy of the foregoing by electronic mail on the following counsel:

Erika H. Arner  
Joshua L. Goldberg  
Christopher C. Johns  
Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P.  
erika.arner@finnegan.com  
joshua.goldberg@finnegan.com  
christopher.johns@finnegan.com

Dated: March 19, 2018

/John M. Caracappa/  
John M. Caracappa  
Reg. No. 43,532

Attorney for Patent Owner  
Broadcom Corporation