

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

HP INC., DELL TECHNOLOGIES INC., DELL INC.,
LENOVO (UNITED STATES) INC.,
Petitioners,

v.

UNIVERSAL CONNECTIVITY TECHNOLOGIES, INC.,
Patent Owner.

IPR2024-01429
U.S. Patent No. 7,187,307

**PETITION FOR *INTER PARTES* REVIEW
UNDER 35 U.S.C. § 312 AND 37 C.F.R. § 42.104**

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	GROUND FOR STANDING.....	1
III.	NOTE.....	1
IV.	SUMMARY OF THE '307 PATENT	1
V.	PROSECUTION HISTORY	4
VI.	LEVEL OF ORDINARY SKILL IN THE ART.....	5
VII.	CLAIM CONSTRUCTION	5
VIII.	RELIEF REQUESTED AND THE REASONS FOR THE REQUESTED RELIEF.....	6
IX.	UNPATENTABILITY OF THE CHALLENGED CLAIMS	6
A.	The Challenged Claims	6
B.	Statutory Grounds for the Claims.....	6
C.	Grounds 1 and 2: Claims 1, 2, 6, 7, and 53 are Anticipated by Shin and Claims 1, 2, 6, 7, and 53 are Obvious over Shin.....	7
1.	Summary of Shin	7
2.	Claim 1	11
3.	Claim 2	27
4.	Claim 6.....	31
5.	Claim 7	32
6.	Claim 53	35
D.	Ground 3: Claims 1, 2, 6, 7, 23, 25, 31, 53, 68, 70, 71, 72, 74, and 75 are Obvious over Yusairi in view of Shin.....	37
1.	Summary of Yusairi.....	37
2.	Summary of Shin	38
3.	Claim 1	38
4.	Claim 2	50
5.	Claim 6.....	52
6.	Claim 7	53

7.	Claim 23	53
8.	Claim 25	57
9.	Claim 31	58
10.	Claim 53	60
11.	Claim 68	61
12.	Claim 70	70
13.	Claim 71	71
14.	Claim 72	71
15.	Claim 74	72
16.	Claim 75	73
E.	CONCLUSION	74
X.	MANDATORY NOTICES	75
A.	Real Party-in-Interest	75
B.	Related Matters.....	75
C.	Lead and Back-up Counsel and Service Information	76
	CERTIFICATE OF WORD COUNT	79
	CERTIFICATE OF SERVICE	80

PETITIONERS' EXHIBIT LIST

Ex. 1001	U.S. Patent No. 7,187,307 (“’307 Patent”)
Ex. 1002	Prosecution History of U.S. Patent No. 7,187,307 (U.S. Patent App. No. 10/459,989)
Ex. 1003	Declaration of Dr. Andrew Wolfe
Ex. 1004	U.S. Patent Pub. No. 2002/0159450 to Shin <i>et al.</i> (“Shin”)
Ex. 1005	“Hardware Design and Implementation of IP-over-1394 Protocol Stack and Its Evaluation,” Mohd Yusairi, Bin Abu Hassan, and Koki Abe, ITE Technical Report Vol.27, No.18, PP 51-85 (March 2003) (“Yusairi”).
Ex. 1006	Declaration of Nathaniel E. Frank-White
Ex. 1007	Declaration of Yukio Iwahana
Ex. 1008	“IEEE Standard for a High Performance Serial Bus,” in IEEE Std 1394-1995, pp.1-384 (1996) (“IEEE1394”)
Ex. 1009	“Serial ATA: High Speed Serialized AT Attachment,” Revision 1.0a, January 7, 2003. (“SATA 1.0a”)
Ex. 1010	“High-Definition Multimedia Interface Specification Version 1.0,” December 9, 2002 (“HDMI 1.0”)
Ex. 1011	“Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications,” IEEE Standard 802.3, 1998 Edition (“IEEE 802.3”)
Ex. 1012	Declaration of Dr. Knut Grimsrud
Ex. 1013	Declaration of Steve Venuti

Ex. 1014	U.S. Pat. No. 6,324,602 to Chen et al. (“Chen”)
Ex. 1015	CV of Andrew Wolfe, Ph.D.
Ex. 1016	English translation, certification, and Japanese original of Apostille from National Diet Library (Tokyo, Japan) regarding Yusairi
Ex. 1017	English translation, certification, and Japanese original of Yusairi from National Diet Library (Tokyo, Japan)
Ex. 1018	“Line Coding,” from EEWeb, https://www.eeweb.com/line-coding/#:~:text=Line%20coding%20is%20the%20process%20of
Ex. 1019	“Line code,” from Wikipedia, https://en.wikipedia.org/wiki/Line_code
Ex. 1020	“8b/10b encoding,” from Wikipedia, https://en.wikipedia.org/wiki/8b/10b_encoding
Ex. 1021	“Duplex (telecommunications),” from Wikipedia, https://en.wikipedia.org/wiki/Duplex_(telecommunications)
Ex. 1022	“Plesiochronous system,” from Wikipedia, https://en.wikipedia.org/wiki/Plesiochronous_system
Ex.1023	“Encapsulation (networking),” from Wikipedia, https://en.wikipedia.org/wiki/Encapsulation_(networking)
Ex. 1024	“Non-return-to-zero,” from Wikipedia, https://en.wikipedia.org/wiki/Non-return-to-zero#:~:text=In%20telecommunications%2C%20a%20non%2Dreturn,other%20neutral%20or%20rest%20condition.

I. INTRODUCTION

HP Inc., Dell Technologies Inc., Dell Inc., and Lenovo (United States) Inc. (“Petitioners”) respectfully request that the Board institute *inter partes* review of and cancel claims 1, 2, 6, 7, 23, 25, 31, 53, 68, 70, 71, 72, 74, 75 (“Challenged Claims”) of U.S. Patent No 7,187,307 (“’307 Patent,” Ex.1001).

II. GROUNDS FOR STANDING

Petitioners certify that the ’307 Patent is eligible for IPR and that Petitioners are not barred or estopped from requesting IPR. 37 C.F.R. § 42.104(a).

III. NOTE

Petitioners cite to the original page numbers of exhibits, rather than branded page numbers. **Emphasis** in quoted material has been added. Claim terms are presented in *italics*.

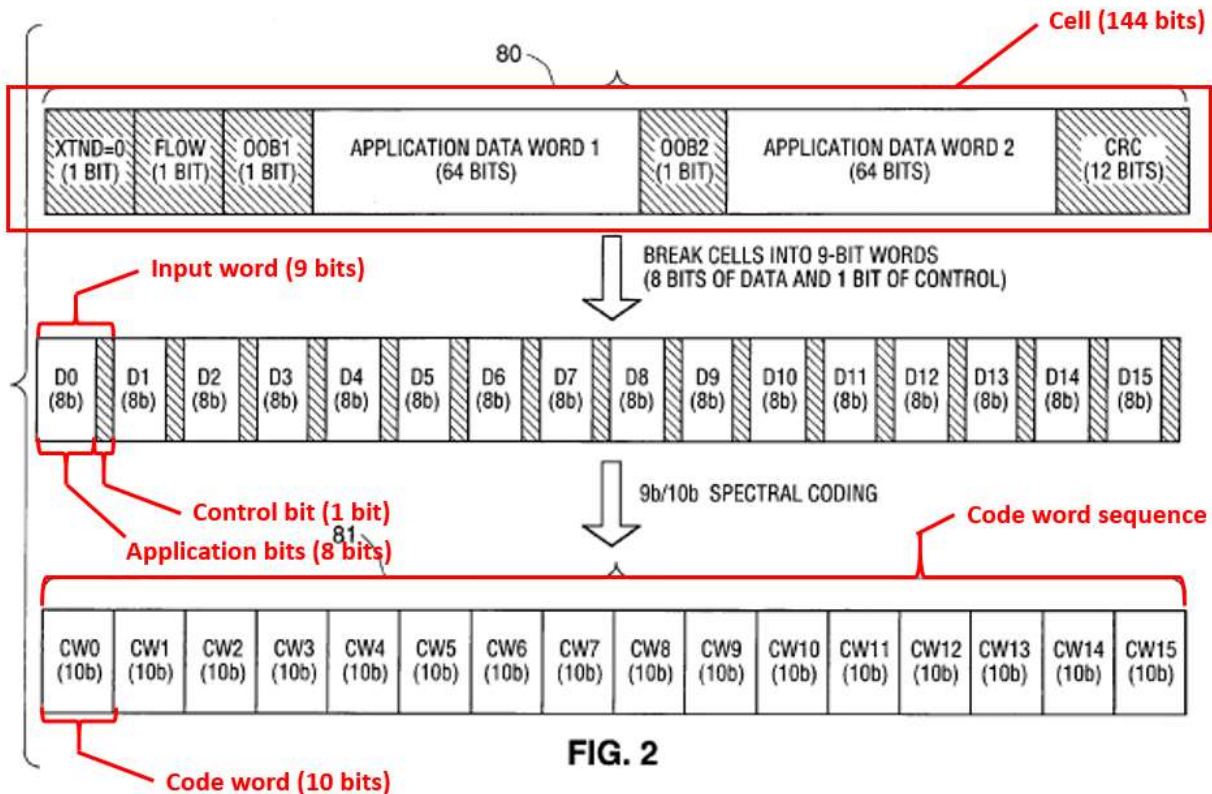
IV. SUMMARY OF THE ’307 PATENT

The ’307 Patent relates to a communication system with at least one transmitter and one receiver. The transmitter is configured to generate encoded data in accordance with a line code for transmission over a serial link. Ex.1001, Abstract. The line code specifies a block code for encoding application data and control bits, and the block code can encode user data bits as a larger number of bits. *Id.* The receiver is configured to decode the encoded data to recover the application data and control bits. *Id.*, Abstract, 3:6-52, 8:39-42. The application data can include data

such as video or audio data. *Id.*, 6:36-38. Control bits cause an endpoint, e.g., the transmitter or receiver, to perform a function. *Id.*, 4:63-66, 6:39-43.

The transmitter and receiver operate according to a stacked architecture, such as the OSI reference model. *Id.*, 1:24-43, 10:20-28, 12:42-50. The OSI model has seven layers: (1) physical layer; (2) link layer; (3) network layer; (4) transport layer; (5) session layer; (6) presentation layer; and (7) application layer. *See id.* With respect to the '307 Patent, the physical and link layers are responsible for transmitting and receiving the encoded line code data. *Id.*, 8:24-25. Any layer above the link layer is referred to as “higher level circuitry,” and the control bits have multiple levels of communication protocol functionality, which makes them functional at, for example, the link layer and a higher layer. *Id.*, 4:63-66, 5:1-14, 6:39-43, 54:62-67.

An exemplary “cell,” along with “input words” and encoded “code words,” is shown below in Figure 2. *Id.*, 4:38, 13:57-61. To encode the data, a cell of 144 bits can be grouped into 16, nine-bit input words, each having eight application bits and one control bit. *Id.*, 14:26-31. The transmitter then applies “block coding” to encode each “input word” as a “code word.” *Id.*, 14:48-52. In the present example, the nine-bit input words are encoded into ten-bit code words. *Id.*, 14:48-52. The code words form a “code word sequence,” which the patent describes as indicative of at least one cell of the input words. *Id.*, 14:48-56.



This block encoding scheme encodes M-bit input words into N-bit code words where $N > M$. *Id.*, 9:19-36; 4:53-59. One example of the scheme is the conventional 8B/10B block code where $M=8$ and $N=10$ such that $N > M$. *Id.*, 3:62-64, 56:10-14. Another example is the 9B/10B block code. *Id.*, 11:45-47. After the transmitter encodes and transmits the line code, the receiver decodes the code words into application data and control bits at its physical layer. *Id.*, 4:56-66. For instance, referring back to Figure 2, the 10-bit code words are decoded into nine-bit words each having eight application bits and one control bit. *Id.*, 11:16-20.

The '307 Patent also suggests that the transmitter can transmit special characters, which “are typically transmitted between cells” and are “distinguishable

from bit sequences of encoded cells.” *Id.*, 5:17-20, 8:45-48. That is, “no special character matches any of the code words.” *Id.*, 5:22-23, 10:55-60. The patent describes “idle characters,” “link training characters,” or “flow control characters” as examples of “special characters.” *Id.*, 9:19-22, 9:33-37, 51:2-4. Idle characters can be inserted between code word sequences for clock synchronization. *Id.*, 49:17-22. Link training characters can be inserted after a link error or startup to establish symbol framing. *Id.*, 17:1-8. Flow control characters can be used to help ensure transmitters and receivers will not overflow their buffers. *Id.*, 53:34-43.

V. PROSECUTION HISTORY

The application that matured into the ’307 Patent was filed on June 12, 2003. The Examiner’s initial search cited only two references. *See* Ex.1002, 187. Examiner then issued a three-page office action that only rejected claim 84 under 35 U.S.C. § 103 in view of one reference—Chen (Ex.1014). Ex.1002, 183-184. The Examiner found that Chen’s “three or more voltage levels for signaling” taught at least two control bits having multiple levels of functionality, and the Examiner concluded that $N > M$ block encoding would be obvious, even though Chen did not specifically disclose it. *Id.*, 184-85.

In response, the Applicant argued that Chen fails to teach: (1) “encoding M -bit words in accordance with a block code to generate a code word sequence of N -bit code words, wherein $N > M$ ” because Chen allegedly taught reducing the voltage

levels relative to the bits, and (2) the limitation requiring “two control bits having multiple levels of communication protocol functionality” because the multiple-level voltage purportedly represented a single, non-binary bit rather than “at least two control bits having multiple levels of communication protocol functionality.” *Id.*, 173.

The Examiner then allowed all 109 claims, providing no reasons for allowance. *Id.*, 137.

VI. LEVEL OF ORDINARY SKILL IN THE ART

A person of ordinary skill in the art (“POSITA”) of the ’307 Patent as of June 2003 would have had a bachelor’s degree in electrical engineering, computer engineering, computer science, or equivalent training, and three years of experience with data communications systems or interfaces, including at least some experience with serial or display data communications systems or interfaces. Additional education can substitute for less work experience, and vice versa. Ex.1003, ¶¶26-29.

VII. CLAIM CONSTRUCTION

Petitioners submit that no terms require construction for this proceeding but reserve the right to propose a construction if the need arises, such as if Patent Owner proposes a construction.

VIII. RELIEF REQUESTED AND THE REASONS FOR THE REQUESTED RELIEF

Petitioners ask that the Board institute *inter partes* review and cancel the Challenged Claims in view of the analysis below.

IX. UNPATENTABILITY OF THE CHALLENGED CLAIMS

A. The Challenged Claims

Petitioners challenge claims 1, 2, 6, 7, 23, 25, 31, 53, 68, 70, 71, 72, 74, and 75.

B. Statutory Grounds for the Claims

Grounds	Claim(s)	Basis
#1	1, 2, 6, 7, 53	35 U.S.C. § 102; these claims are anticipated by Shin
#2	1, 2, 6, 7, 53	35 U.S.C. § 103; these claims are obvious over Shin
#3	1, 2, 6, 7, 23, 25, 31, 53, 68, 70, 71, 72, 74, 75	35 U.S.C. § 103; these claims are obvious over Yusairi in view of Shin

U.S. Patent Application Publication No. 2002/0159450 to Shin et al. (“**Shin**,” Ex.1004) was filed November 7, 2001, and published October 31, 2002. Shin is prior art at least as of November 7, 2001 under pre-AIA 35 U.S.C. § 102(e), and at least as of October 31, 2002, under pre-AIA 35 U.S.C. § 102(a).

The article entitled “Hardware Design and Implementation of IP-over-1394 Protocol Stack and Its Evaluation,” by Mohd Yusairi et al., ITE Technical Report

Vol.27, No. 18, pp. 51-85 (March 2003) (“**Yusairi**,” Ex.1005), is a printed publication that was published on March 4, 2003, and made publicly accessible by no later than April 22, 2003. *See* Ex.1007; Ex.1016 (Apostille from National Diet Library with certified translation); Ex.1017 (Yusairi original with certified translation). Thus, Yusairi is prior art as of at least April 22, 2003, under at least pre-AIA 35 U.S.C. § 102(a).

None of the prior art was cited during prosecution of the ’307 Patent.

C. Grounds 1 and 2: Claims 1, 2, 6, 7, and 53 are Anticipated by Shin and Claims 1, 2, 6, 7, and 53 are Obvious over Shin

1. Summary of Shin

Shin discloses a serial data communications architecture for communication between hosts and data store devices. Ex.1004, Abstract, [0070]. The Shin architecture is a stacked communication protocol architecture, including “transport, link, and physical layers.” *Id.*, [0071]. “[E]ach layer in the hierarchy is aware of and can take advantage of characteristics of lower layers in the hierarchy.” *Id.*, [0071]. For example, the transport layer “may receive signals from the link layer that indicate when an end of packet is received.” *Id.*, [0071]. The link layer can also process control information for functions at the link layer. *Id.*, [0096] (discussing synchronization primitives).

Shin's architecture is depicted in Figure 1 below. Shin includes hosts 110 and data store devices 120 interconnected by a switching network 130 via serial communications link 140. *Id.*, [0075], Fig. 1. Each of the hosts 110, data store devices 120, and switching network 130 includes a transport layer, link layer, and physical layer:

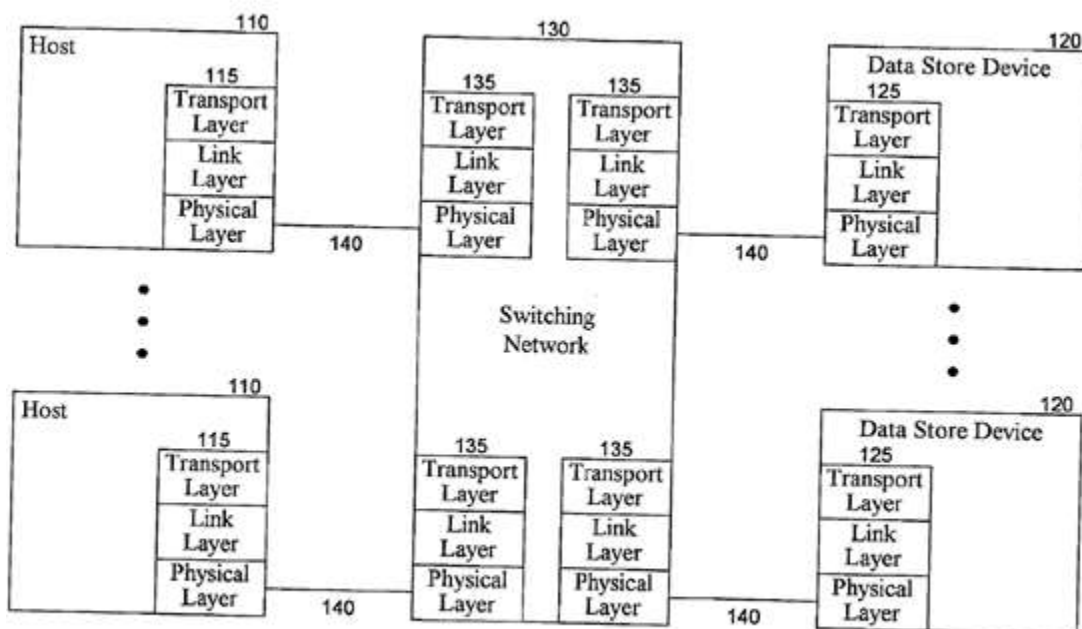
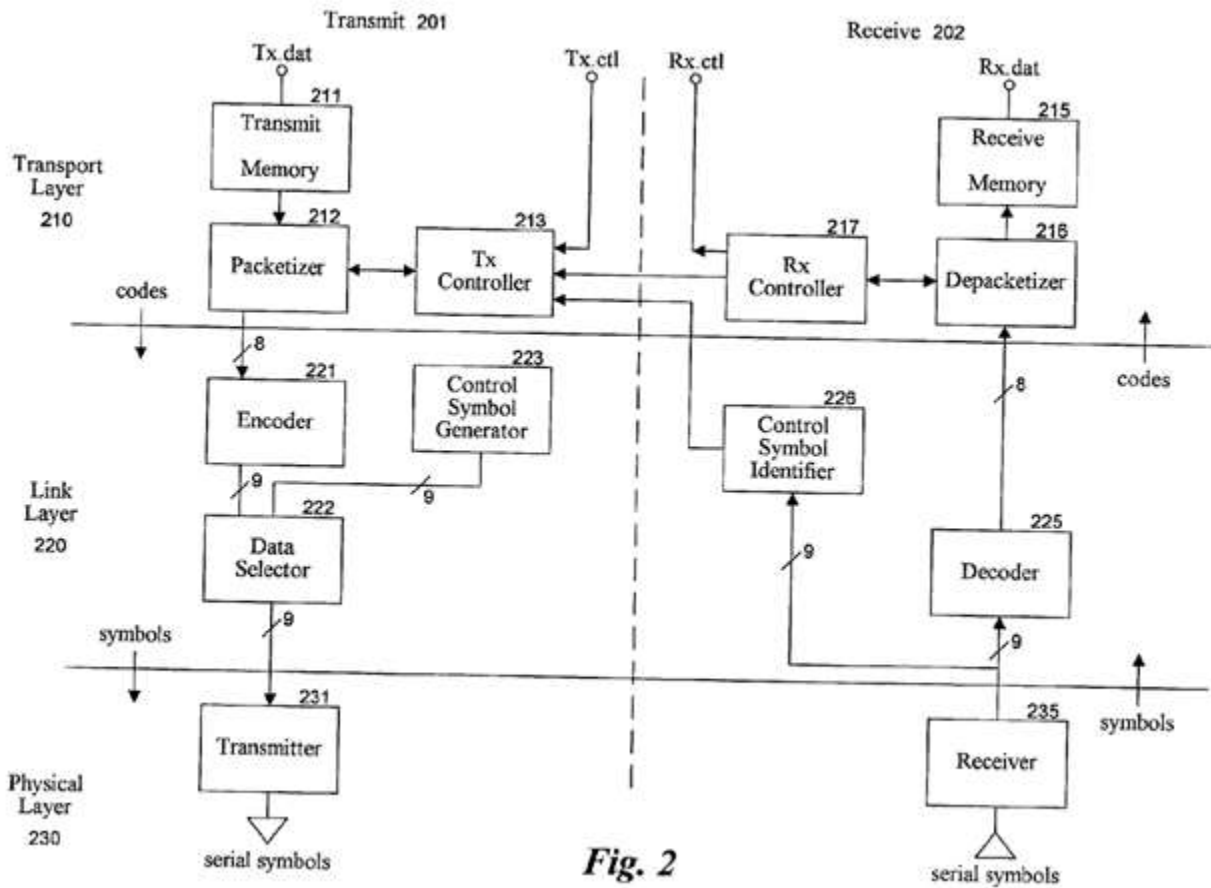


Fig. 1

Id., Fig. 1.

The transmit components of the link layer include an encoder 221, a data selector 222, and a control symbol generator 223, and the receive components of the link layer include a decoder 225 and a control symbol identifier 226, as shown below in Figure 2 of Shin. *Id.*, [0079].



Id., Fig. 2. The data selector 222 “inserts control symbols and synchronization symbols into the sequence of symbols that are to be transmitted.” *Id.*, [0079]. On the receive side 202, the decoder 225 of the link layer decodes symbols into codes, and the control symbol identifier 226 identifies control symbols and signals the transport layer as appropriate. *Id.*

For example, Shin describes synchronization primitives that the transmitter periodically transmits to align symbols at the receiver and that may also encode

packet type.¹ *Id.*, [0096]. As shown in Figure 9A below, a synchronization primitive 901 encoded with a packet type of “control” is transmitted immediately prior to a control packet 902, and a synchronization primitive 905 encoded with a packet type of “data” is transmitted immediately prior to a data packet 907.²

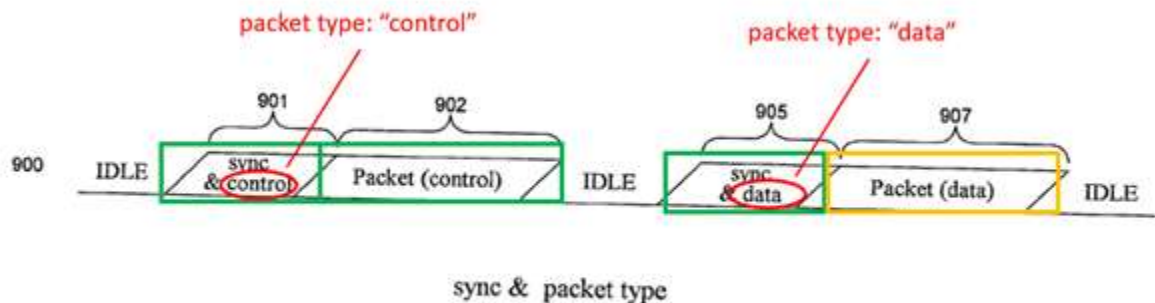


Fig. 9A

The link layer of the receiving node can detect the packet type of the received packets by decoding the synchronization primitive. Ex.1004, [0096].

¹ Shin describes that “primitives are pairs of out-of-band symbols” and states that “a symbol is assumed to have nine bits.” Ex.1004, [0098], [0138]-[0139].

² Shin’s specification refers to a “synchronization primitive 903” encoded with a packet type of “data” and “a data packet 904,” but based on the specification as a whole, a POSITA would have understood that these refer to reference numerals 905 and 907, respectively. Ex.1003, at 24, FN2.

Shin states that the link layer can also detect a Preempt primitive to interrupt transmission of a packet, as shown in Figure 13 below. *Id.*, [0111]-[0115].

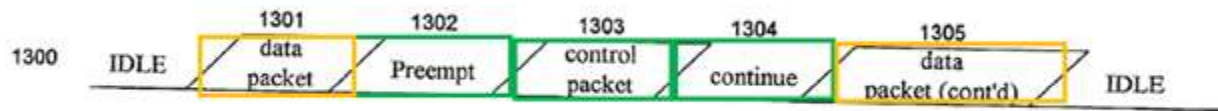


Fig. 13

Id., Fig. 13 (annotated). The link layer of the receiving node detects the Preempt primitive and signals to the node's transport layer, which then performs the function of "sav[ing] current packet information" for the data packet "so that it can continue storing codes in the [data] packet after the preemption is complete." *Id.*, [0115]. The transport layer "starts storing the subsequent codes as a control packet" (the preempting packet) until the link layer indicates it has received a continue primitive for resuming the data packet. *Id.*, [0115], [0112].

2. Claim 1

[1(pre)]³ A communication system, including:

To the extent the preamble is limiting, Shin discloses this limitation. As provided below, Figure 1 illustrates a *communication system* that includes a storage

³ These claim limitation designations are set forth in the Claim Listing Appendix at the end of this Petition.

link architecture having hosts and data store devices that communicate with each other. *Id.*, [0075], FIG. 1; Ex.1003, ¶¶61-62.

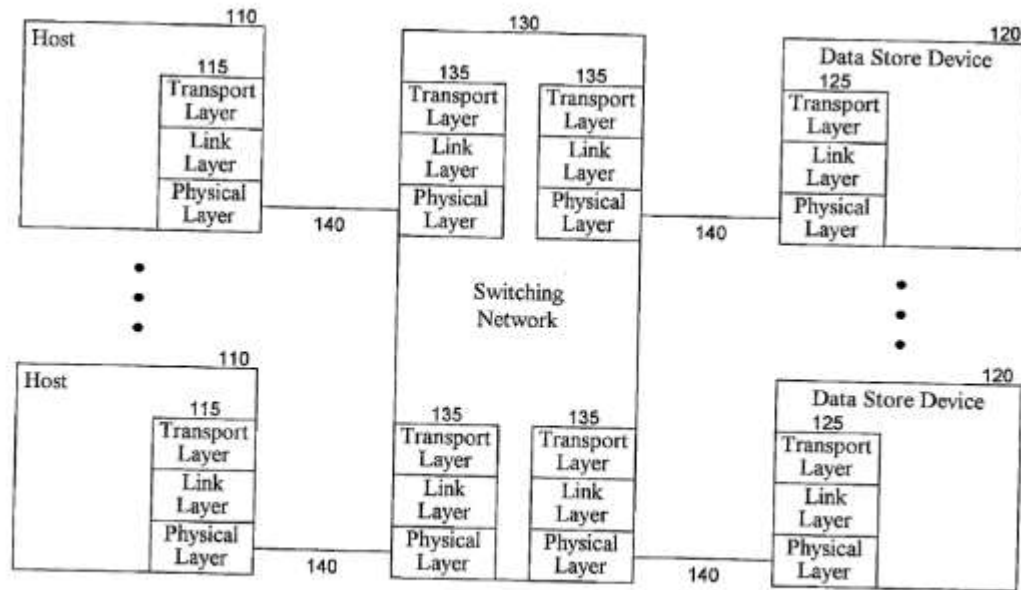


Fig. 1

[1(a)] a first endpoint;

Shin discloses this limitation. Shin's storage link architecture supports packet-based communications between hosts and storage devices, where a host acts as a transmitter (*a first endpoint*) to transmit data to a data store device. Ex.1004 at [0071], [0076], [0077]; Ex.1003, ¶63. Annotated Figure 1 below depicts a host (**purple**) acting as a *first endpoint*.

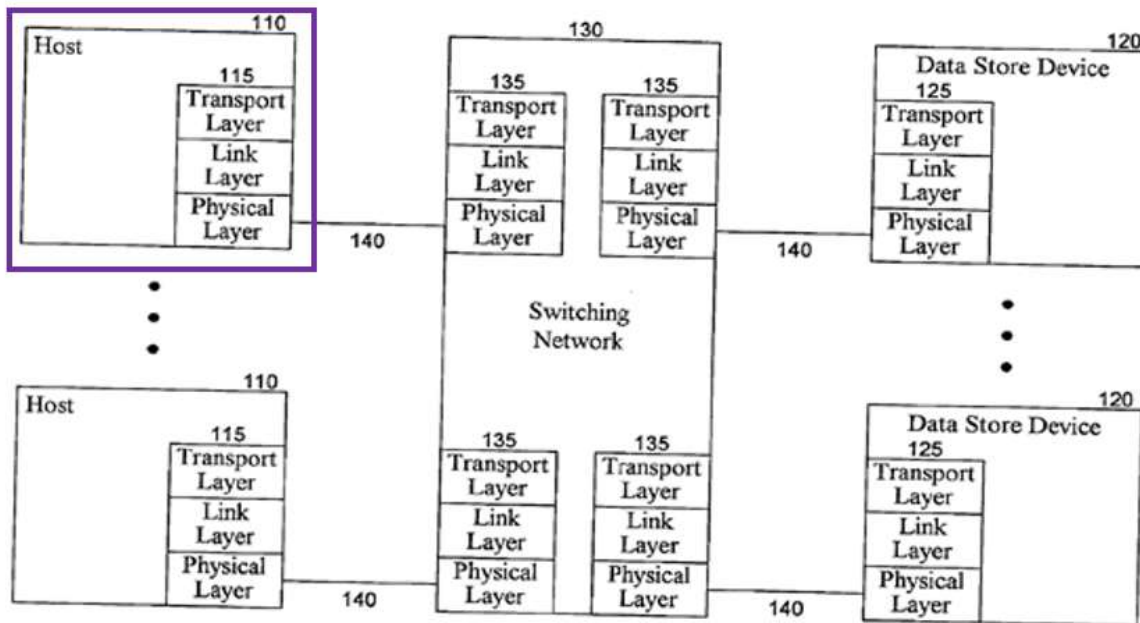


Fig. 1

[1(b)] a second endpoint, including physical and link layer circuitry and higher level circuitry coupled to the physical and link layer circuitry; and

Shin discloses this limitation. “Each host, data store device, and switch is a communications node that includes one or more communications interfaces 115, 125 and 135 with a transport layer, a link layer, and a physical layer.” Ex.1004 at [0076], FIG. 1. A host communicates to a data store device 120 (*the second endpoint*) (**blue**). *Id.*, FIG. 1. As illustrated, the data store device acts as the second endpoint and includes a transport layer (*transport layer circuitry*) (**red**) and link layer (*link layer circuitry*) and physical layer (*physical layer circuitry*) (**green**). *Id.*; Ex.1003, ¶¶64-65.

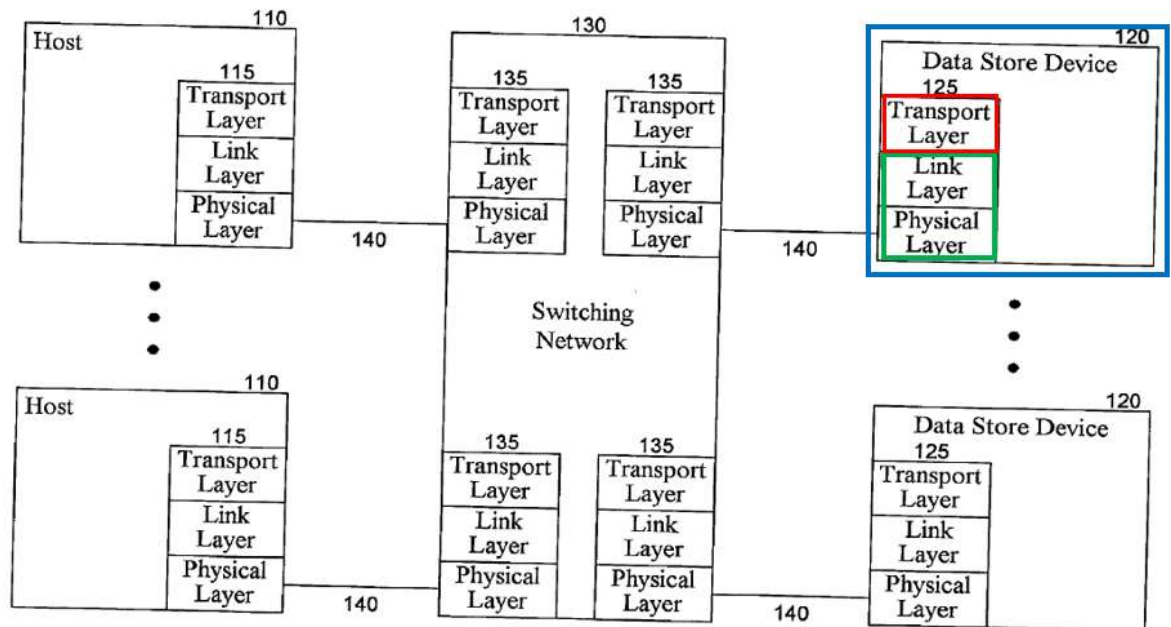


Fig. 1

The transport layer (*higher level circuitry*) is coupled to the physical and link layer circuitry so that “[t]he transport layer may indicate to the link layer when to transmit control symbols, and the link layer may indicate to the transport layer when control symbols are received.” Ex.1004 at [0079]; Ex.1003, ¶66.

[1(c)] a serial link between the first endpoint and the second endpoint,

Shin discloses this limitation. Shin provides that a host (*the first endpoint*) and a data store device (*the second endpoint*) “are interconnected to switching network 130 via serial communication links 140.” Ex.1004 at [0075]. Figure 1 below illustrates a serial communication link (*a serial link*) (blue) between the host (*the first endpoint*) and the data store device (*the second endpoint*). *Id.*; Ex.1003, ¶¶67-69.

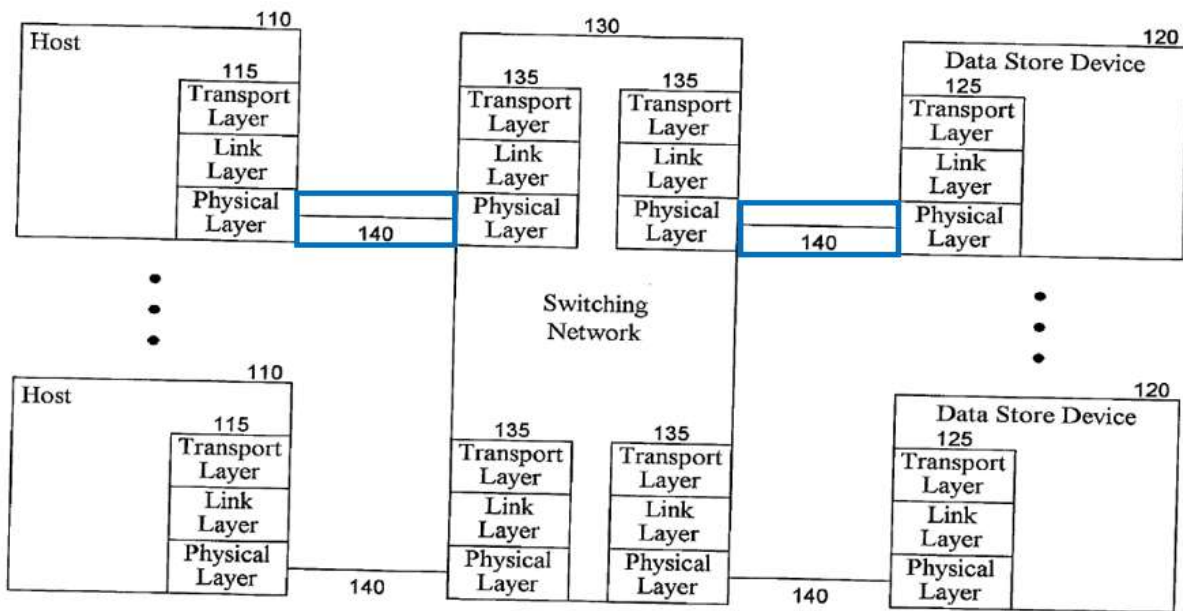


Fig. 1

[1(d)] wherein the first endpoint is configured to generate encoded data in accordance with a line code, and to transmit the encoded data over the link to the second endpoint, and

Shin discloses this limitation. Shin teaches that “each host [*the first endpoint*], data store device [*the second endpoint*], and switch is a communications node.” Ex.1004 at [0076]. Shin also teaches that a communications node (a host for example) “encodes codes into symbols in order to optimize transition control and to ensure DC-balance.” *Id.*, [0128], Ex.1003, ¶¶70-71.

A “line code” would have been understood to include digitally encoding binary data to be transmitted over a line. Ex.1003, ¶73. A POSITA would have understood that Shin discloses digitally encoding binary data to be transmitted over

a line (e.g., link 140), and, therefore, discloses *generat[ing] encoded data in accordance with a line code*. Ex.1003, ¶73.

Further, the '307 Patent describes encoding data bits into larger groups or sequences of bits (i.e., blocks) using block code and transmitting them as “line code [that] specifies [the] block code” so as to “ensure high transition density to facilitate clock/data recovery, and accomplish DC balancing.” Ex.1001, 1:34-44, Abstract, 4:53-56, 5:15-18; Ex.1003, ¶72. Consistent with this, Shin discloses using 8B/9B block encoding to encode 8-bit data into 9-bit symbols (i.e., into a larger number of bits), Ex.1004 at [0131], and transmitting them to “optimize transition control,” maintain clock recovery and error detection, and “ensure DC-balance.” *Id.*, Abstract, [0071], [0077], [0123]-[0124], [0128]; Ex.1003, ¶72-74. As such, a POSITA would have understood the Shin encodes data *in accordance with a line code*. Ex.1003, ¶75.

Finally, Shin discloses an example of a line encoding technique in another way. Shin teaches using selective block inversion to transition optimize a block code. Ex.1004 at [0131]-[0132] (“The encoding technique may also help ensure DC-balance by using selective block inversion encoding on symbols that have been transition optimized”). A POSITA would have understood that selectively inverting encoded symbols to generate DC-balance requires a line code. *Id.*, [0132]; Ex.1003, ¶75. Thus, Shin’s host transmitter (*the first endpoint*) is configured to generate

encoded data in accordance with a line code (at least by selectively inverting 9-bit symbols), and to transmit the encoded data (the selectively inverted symbols) over the link (the serial communication link) to the second endpoint (the data store device). Ex.1003, ¶75.

[1(e)] wherein the physical and link layer circuitry of the second endpoint is coupled to the link and configured to decode the encoded data

Shin discloses this limitation. “The physical layer [of *the second endpoint*] receives serialized symbols via the communications link [*the link*], deserializes the symbols, and provides the symbols to the link layer.” Ex.1004, [0077]. At the link layer, “[t]he data decoder receives [the] symbols from the physical layer and decodes them.” *Id.*, [0079], Fig. 1. Thus, Shin discloses *physical and link layer circuitry of the second endpoint is coupled to the link and configured to decode the encoded data*. Ex.1003, ¶¶76-78.

[1(f)] to recover application data and control bits,

Shin discloses this limitation. Shin discloses that its transmitting node (e.g., a host) transmits “data packets” that include “blocks of data to be stored” at the receiving node (e.g., a data store device). Ex.1004, [0090], [0088], Fig. 5, [0093]-[0094], Figs. 7-8. In addition to data packets, the host also transmits primitives, such as synchronization primitives that encode packet type, which may be transmitted immediately prior to packets to designate their packet type. *Id.*, [0097]. As shown

in the annotated version of Figure 9A below, a synchronization primitive 901 encoded with a packet type of “control” is transmitted immediately prior to a control packet 902, and a synchronization primitive 905 encoded with a packet type of “data” is transmitted immediately prior to a data packet 907. Ex.1003, ¶¶79-80.

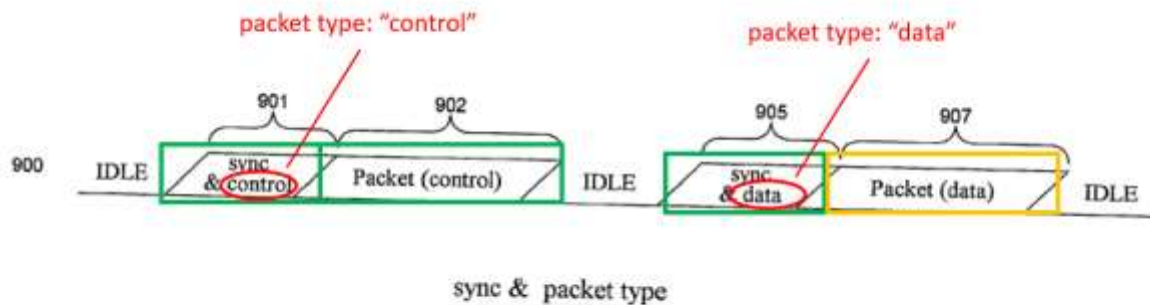


Fig. 9A

Ex.1004 at Fig. 9A (annotated), [0097]. “The receiving communications node uses the synchronization primitive encoded with packet type to align its symbol boundaries and to rapidly identify the packet type.” *Id.*, [0097].

Similarly, Shin discloses transmitting a control primitive in the form of a Preempt primitive to interrupt the transmission of a packet, as shown below. *Id.*, [0111]-[0115].

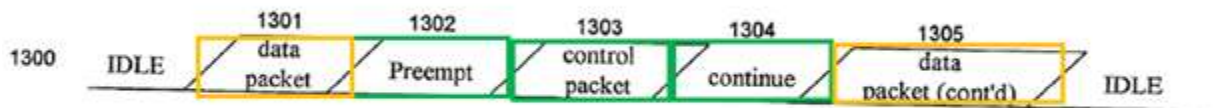


Fig. 13

Id., Fig. 13 (annotated).

As shown in the annotated version of Figure 9C below, a link layer 910 of a transmitter is provided with bits of the code (labeled “Data”) (orange) that is encoded to be transmitted and provided with an indication of a primitive (labeled “CTL_S”) (green) for transmission. *Id.*, [0105]. Ex.1003, ¶80.

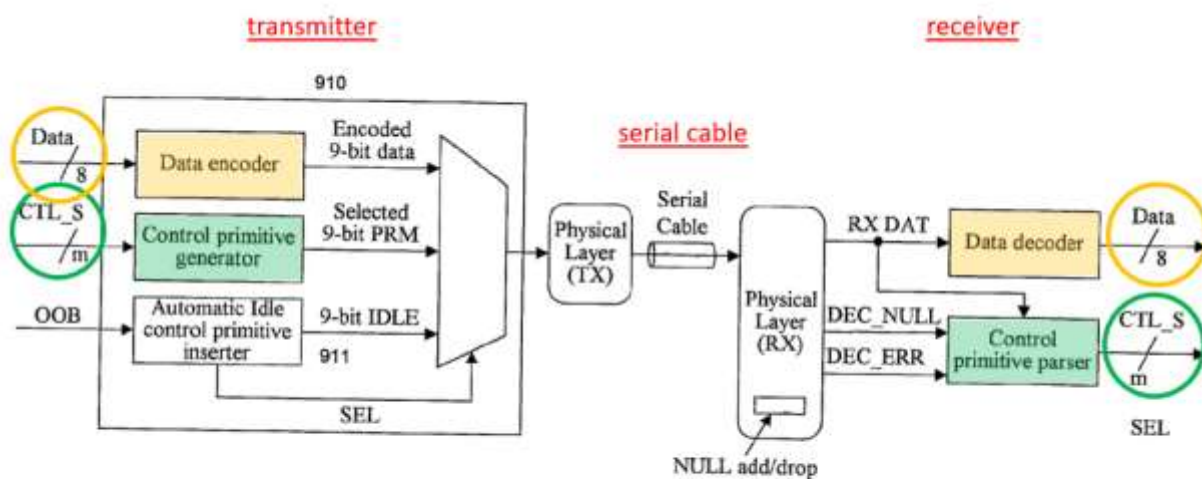


Fig. 9C

Id., Fig. 9C. As shown, after receipt, the receiver data (RX DAT) that includes the encoded data is decoded by the data decoder (orange) to recover bits of data (e.g., 8 bits), while the primitives of the receiver data (RX DAT) are parsed by the control primitive parser (green) to recover “m” control bits. *Id.*, Fig. 9C, [0077] (describing decoding at the receiver), [0079] (same), [0146] (describing decoding primitives), Fig. 29; Ex.1003, ¶80. Preempt primitives are an example of such “control primitives” from which “m” control bits are recovered. Ex.1004, [0112]. Ex.1003,

¶81. Further, Shin discloses that “control bit[s]” (e.g., “[V_n, V_{n+1}]”) (Ex.1004, Figs. 40-41) may be recovered from a synchronization primitive, as depicted in Figures 38A, 39B, 40, and 41, and described in [0163], [0165], [0168], and [0170]-[0171]. Ex.1003, ¶82-85.

The decoded “Data” that is transmitted via “data packets” (**orange**) constitutes *recover[ed] application data*, and the “m” control bits recovered from parsing control primitives (“CTL_S”) (**green**), such as preempt primitives, as well as the “control bit[s]” recovered from synchronization primitives, constitute *recover[ed] control bits*. Ex.1004, [0098]; Ex.1003, ¶82-85. Further, Shin discloses that all of its primitives are comprised of “bits” that are recovered from the primitives when received, and these bits are “control bits” because they cause functions to be performed by layers of the receiver. Ex.1004, [0098], [0096]-[0097], [0115]; Ex.1003, ¶81.

Additionally, a POSITA would have understood that, since the 8-bits of data (**orange**) and the “m” control bits (**green**) and the “control bit[s]” are each recovered by decoding packets and parsing primitives, they are recovered from the line code to allow the digital signal transmitted across the line to be converted back to bits. Ex.1003, ¶87-88.

[1(g)] to perform at least one link level function in response to at least one of the control bits, and

Shin discloses this limitation. Specifically, Shin discloses that the link layer of its receiving node can perform the function of detecting packet type (*perform[ing] at least one link level function*) in response to receiving a synchronization primitive encoded with packet type (*at least one of the control bits*). Ex.1004, [0096]-[0097]; Ex.1003, ¶¶89-90. “The receiving communications node can take the appropriate action based on the packet type encoded in the synchronization primitive....” *Id.*, [0096]. “Also, when a synchronization primitive encodes the packet type, **then the link layer, rather than the transport layer, can detect packet type**, which can enhance the overall processing speed.” *Id.*, [0096].

Because Shin’s receiving node performs the link level function of detecting packet type in response to receiving a synchronization primitive encoded with packet type, this function is performed *in response to at least one of the control bits*. Ex.1003, ¶91.

[1(h)] to assert at least one control signal indicative of at least another one of the control bits to the higher level circuitry, and

Shin discloses and renders obvious this limitation. Shin discloses that when the link layer receives and parses a preempt primitive to recover bits (*another one of the control bits*), the link layer “signals” to the transport layer component that a preempt primitive has been received (*assert at least one control signal indicative of at least another one of the control bits to the higher-level circuitry*). Ex.1004,

[0115]; Ex.1003, ¶¶92-93. Specifically, Shin discloses that “if the link layer **signals** that a **preempt primitive** has been received” (block 1502), then “the component saves current packet information so that it can continue storing codes in the packet after the preemption is complete.” (block 1504). *Id.*, [0115]. Thus, in Shin, the link layer *assert[s] at least one control signal* to the transport layer component (*higher level circuitry*) where the signal is *indicative of at least another one of the control bits*, because the link layer “signals” the transport layer component in response to a preempt primitive, which, as discussed for limitation [1(f)], generates “m” control bits (see Figure 9C) upon receipt and parsing of the primitive. Ex.1003, ¶93. Thus, a POSITA would have understood that the “signal” asserted by the link layer to the transport layer is *indicative of control bits* because the signal occurs in response to receiving the control bits of the preempt primitive. Ex.1003, ¶93. *Id.*, [0115].

A POSITA would have further understood that the control bits recovered from the preempt primitive would constitute *at least another one of the control bits*, at least to the extent the term “another one of the control bits” encompasses one or more different control bit values and does not require different control bit fields. Ex.1003, ¶94. This is because a POSITA would have understood that the preempt and synchronization primitives generate different control bit values when received because they have different functionality. Ex.1003, ¶94.

To the extent the term “*at least another one of the control bits*” requires different control bit fields from “*at least one of the control bits*,” a POSITA would have found it obvious and would have been motivated to implement Shin’s synchronization primitive and the preempt primitive using different control bit fields. Ex.1003, ¶95. First, a POSITA would have known that there would have been a finite number of foreseeable solutions—either i) encode the primitives using separate control bit values in common bit fields, or ii) encode the primitives using separate bit fields. Ex.1003, ¶95. This would have amounted to a routine design choice between two alternatives that would have been obvious to a POSITA. Ex.1003, ¶95. Second, encoding control primitives using separate bit fields was a well-known solution in the field at the time, as demonstrated by, for example, the SATA 1.0a standard that had been adopted and publicly released in early 2003, prior to the filing of the ’307 Patent, and that was intended to provide serial connections to storage devices (like Shin). Ex.1003, ¶96; Ex.1009, 1, 143-144, FIG. 57; Ex.1012. SATA demonstrates that it was well known to encode control primitives by using separate bit fields because it teaches using primitives both within and outside of frames and including primitives in dedicated bit fields, such as the SOF primitive that starts a frame, and the EOF primitive that always ends a frame (both in **orange**). Ex.1009, 143-144, 146-147, FIG. 57; Ex.1003, ¶96.

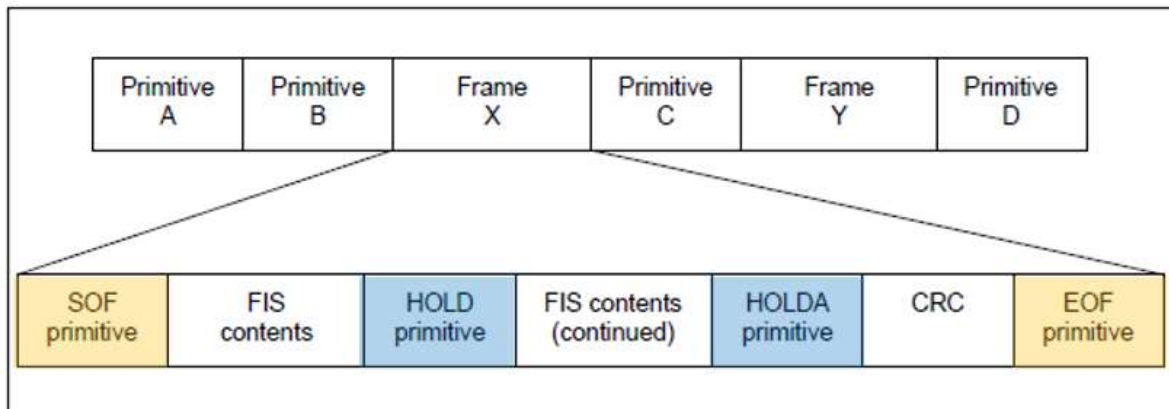


Figure 57 – Transmission structures

Ex.1009, FIG. 57 (annotated), 144; Ex.1003, ¶96 (discussing how separate bit fields were also used for the HOLD and HOLDA primitives (in blue); see also Ex.1012, ¶¶17-20 (discussing separate, dedicated bit fields for SOF, EOF, HOLD, and HOLDA primitives in the SATA 1.0a protocol). That it was well known in this field to encode control primitives using separate bit fields is further demonstrated by the earlier HDMI 1.0 standard, which uses separate bit fields to encode control information, such as the bit fields CTL0, CTL1, CTL2, and CTL3, illustrated below. Ex.1010, 42-43, 43 (Table 5-2); Ex.1013.

Table 5-2 Preambles for Each Data Period Type

CTL0	CTL1	CTL2	CTL3	Data Period Type
1	0	0	0	Video Data Period
1	0	1	0	Data Island Period

Ex.1010, 43 (Table 5-2). For example, CTL2 encoded packet type, while CTL3 encoded encryption status. *Id.*, Table 42-43; Ex.1003, ¶96. Thus, a POSITA would

have been motivated to encode control primitives using separate control bit fields because it was well known in the art. Ex.1003, ¶96.

Third, Shin discloses up to 18 bits available for control bit encoding that can represent up to 256 out-of-band symbols. Ex.1004, [0158]. Thus, a POSITA would have had a reasonable expectation of success encoding the synchronization primitive and preempt primitive using separate control bit fields because this was a well-known technique and Shin provides more than enough control bits allowing such to be implemented. Ex.1003, ¶97.

Therefore, it would have been an obvious and a well-known design choice from among two finite options for a POSITA to encode Shin’s primitives in separate control bit fields, to the extent such is deemed to be required for this limitation [1(h)]. Shin, therefore, discloses and renders obvious this limitation. Ex.1003, ¶98.

[1(i)] wherein the higher level circuitry is configured to perform at least one higher level function in response to the at least one control signal.

Shin discloses this limitation. As discussed for limitation [1(h)], “if the link layer signals” to the component of the transport layer (*higher level circuitry*) “that a preempt primitive has been received” in “decision block 1502,” then “the component continues at block 1504 ... [and] the component saves current packet information [*perform[s] at least one higher level function*] so that it can continue storing codes in the packet after the preemption is complete.” Ex.1004, [0115], [0112], Fig. 15. The

transport layer component then, in block 1505, also “processes the preempting packet,” which “includes receiving each code and storing the codes in a preempting packet until the link layer indicates that it received a continue primitive,” which also constitutes *perform[ing] at least one higher level function in response to the at least one control signal* because it is done in response to the link layer signaling the transport layer that the preempt primitive has been received. *Id.*, [0115], FIG. 15; Ex.1003, ¶¶99-100. Figure 15 of Shin is shown below, with the block 1502 (purple) step of the link layer asserting the control signal to the transport layer, and with the blocks 1504 and 1505 (blue) showing the higher level functions performed by the higher transport layer in response to the control signal:

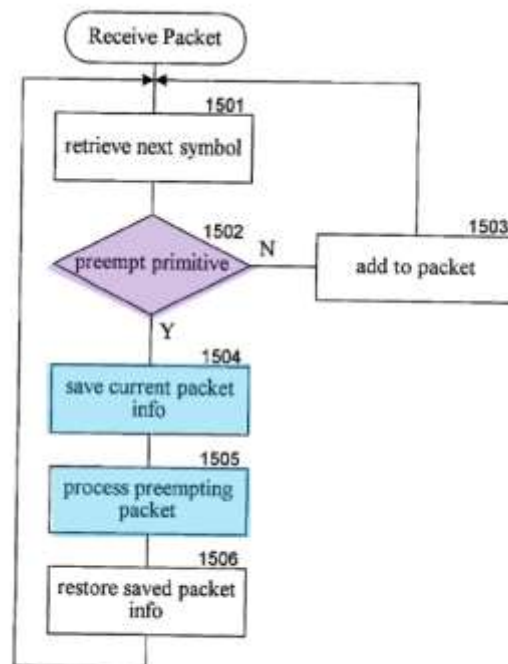


Fig. 15

Ex.1004, Fig. 15 (annotated).

3. Claim 2

[2] The system of claim 1, wherein the line code specifies a block code and special characters, the first endpoint is configured to encode M-bit input words in accordance with the block code to generate a code word sequence of N-bit code words, where $N > M$, the code word sequence is indicative of at least one cell of the input words, and the encoded data are a sequence of at least one of the special characters and the code words of the code word sequence.

Shin discloses and renders obvious this limitation.

As described above with respect to limitation [1(d)], *the line code specifies a block code* because a POSITA would have understood that Shin's 8B/9B uses block encoding to generate block code that is transmitted digitally across a line by line code. Ex.1003, ¶102-103.

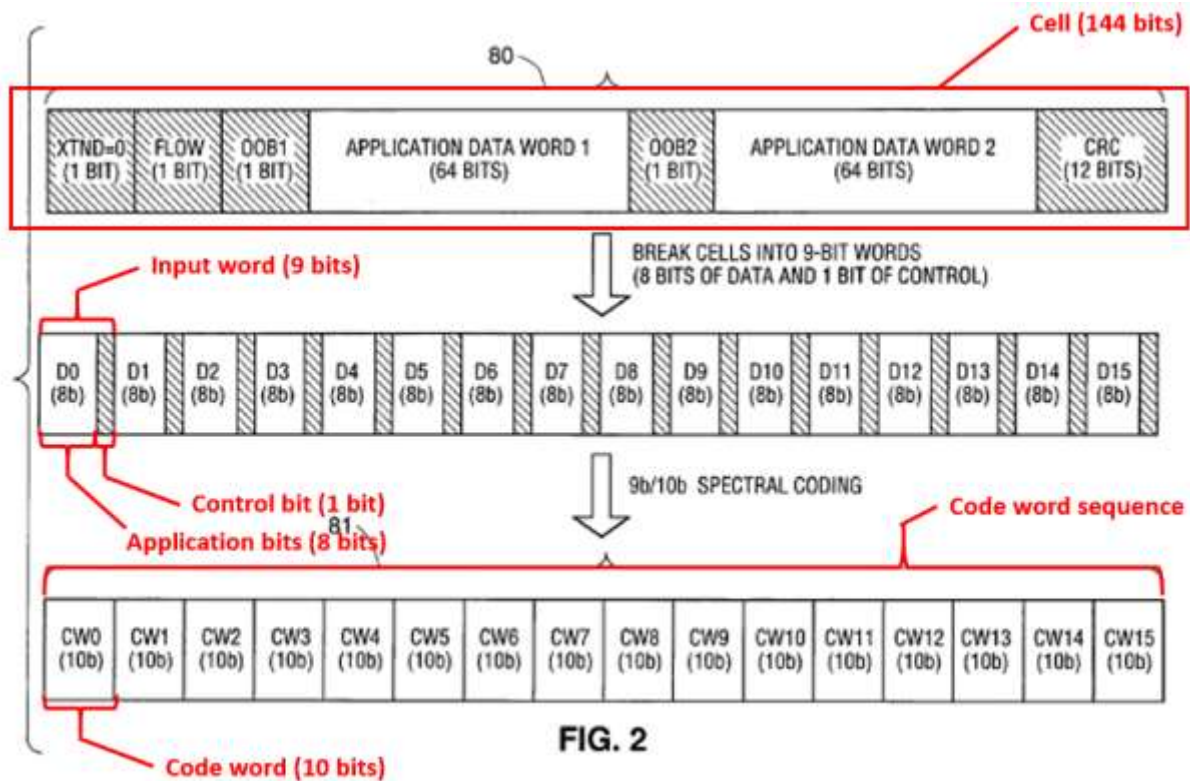
Further, Shin discloses that line code also specifies *special characters*. Shin describes out-of-band symbols that teach the *special characters* because an out-of-band symbol is “inserted into a sequence of in-band symbols” and “does not have the same desirable characteristics as in-band symbols.” Ex.1004, [0086]. Likewise, the *special characters* described by the '307 Patent are “inserted in a stream of code words” and are “distinguishable from ... code word sequences.” Ex.1001, 18:48-50, 27:66-28:2. A POSITA would have understood that Shin's out-of-bound symbols are *special characters* because both are inserted into in-band encoded data blocks and do not have the same characteristics as the in-band encoded data blocks.

Ex.1003, ¶104. A POSITA would have further understood that *the line code specifies ... special characters* because the out-of-band symbols (*special characters*) can be inserted “into the sequence of symbols that are to be transmitted” digitally over the line using the line code. Ex.1004, [0079]; *id.*, [0076] (describing that out-of-band symbols (*special characters*) in the form of control symbols can be inserted and interspersed into a sequence of in-band symbols (block encoded code words), both of which are communicated via a communications link to a another communications node); Ex.1003, ¶103.

Shin also teaches *the first endpoint is configured to encode M-bit input words in accordance with the block code to generate a code word sequence of N-bit code words, where $N > M$* , because, as described with respect to the discussion at limitation [1(d)], Shin teaches an 8B/9B encoding method for *the first endpoint* to encode data. A POSITA would have understood that the 8B/9B encoding scheme used by *the first endpoint* is a block encoding scheme that encodes eight-bit (*M-bit*) input words into *a code word sequence of nine-bit (N-bit) code words in accordance with the 8B/9B encoding (the block code)*. Ex.1003, ¶105. Thus, $N > M$. *Id.*

Shin further teaches that *the code word sequence is indicative of at least one cell of the input words*. According to the '307 Patent, a “cell,” in the context of “packet-based (cell-based) communication,” includes “application and control bits.” Ex.1001, 4:37-38. “Each cell has a predetermined structure, so that the position of a

bit within the cell determines whether the bit is an application data bit or a control bit and determines the type (function) of each control bit.” *Id.*, 4:63-66. Annotated Figure 2 “shows an example of such a cell, consisting of 144 bits.” *Id.*, 14:23-25.



Id., Fig. 2. “The bits of each cell are grouped into input words” having “application data bits ... [and] control bit[s].” *Id.*, 14:26-31. “[I]nput words indicative of cell are encoded ... using a 9-Bit-to-10-Bit (‘9B/10B’) block code to produce the sixteen 10-bit code words (CWO—CW15) shown. The code words are transmitted in an appropriate sequence over a serial link.” *Id.*, 14:48-53; Ex.1003, ¶106.

Likewise, Shin describes a packet (*cell*) that is encoded and transmitted using block code. Ex.1004, [0076]. For example, a “packet includes a header section and

a payload section.” *Id.*; Ex.1003, ¶107. “The transport layer receives the data of the transaction and generates packets (i.e., packetizes the data) for transmission. The transport layer then provides each code (e.g., each byte) of the packets to the link layer.” Ex.1004, [0076]; Ex.1003, ¶108. “The link layer receives codes to be transmitted from the transport layer, encodes the codes into symbols, and provides those symbols to the physical layer.” Ex.1004, [0077]. The link layer can use block encoding to encode the words into symbols, such as an “encoding technique [that] adds a ninth bit ... to the 8-bit code, ... which results in a 9-bit symbol.” *Id.*, [0129]. Ex.1003, ¶109. This generates “a sequence of symbols that are to be transmitted.” Ex.1004, [0079]. “The physical layer receives the symbols from the link layer, serializes the symbols, and transmits the serialized symbols via the communications link.” *Id.*, [0077]. Thus, Shin discloses generating codes (*input words*) from a packet (*cell*) and encoding the codes (*input words*) into symbols (*code words*) using block encoding techniques. As such, Shin’s sequence of symbols (*code word sequence*) is indicative of at least one packet (*cell*) of the codes (*input words*). Ex.1003, ¶109-110.

Lastly, Shin discloses *the encoded data are a sequence of at least one of the special characters and the code words of the code word sequence*. Shin discloses a data selector at the link level that “effectively inserts control symbols [*special characters*] and synchronization symbols into the sequence of symbols [*code word*”

sequence] that are to be transmitted.” Ex.1004, [0079]. As noted, out-of-band symbols are special characters, and Shin also refers to out-of-band symbols as “control symbols.” *Id.*, [0086] (“Because control information is encoded as out-of-band symbols, the control symbols can be inserted into a sequence of in-band symbols and readily detected as control symbols when received.”); Ex.1003, ¶111.

In a specific example, Shin discloses a special character called an “IDLE symbol” (*at least one of the special characters*), also referred to as an “IDLE primitive.” Ex.1004, [0086] (“an IDLE symbol is actually a sequence of two out-of-band symbols”). IDLE symbols (*at least one of the special characters*) can be inserted “after every block of a packet.” *Id.*, [0086]; *id.*, [0105]; Ex.1003, ¶112. Thus, because a control symbol, such as an IDLE symbol, (*at least one of the special characters*) is inserted into the sequence of symbols (*code words of the code word sequence*) for transmission, Shin describes encoded data that is a sequence of at least one special character and the code words of the code word sequence. Ex.1003, ¶112.

4. Claim 6

[6] The system of claim 2, wherein said at least one of the special characters is an idle character.

Shin discloses this limitation. As described above at limitation [2], Shin discloses an example of an out-of-band symbol (*special character*) that is an idle symbol (*idle character*). Ex.1004, [0086]; Ex.1003, ¶113.

5. Claim 7

[7] The system of claim 6, wherein the physical and link layer circuitry is configured to perform simplex link flow control by inserting idle characters between subsequences of the code word sequence, where each of the subsequences is indicative of a different cell of the input words.

Shin discloses and renders obvious this limitation. Shin discloses a receiver (the second endpoint) having a physical layer (*physical and link layer circuitry*) that inserts a NULL primitive (*idle character*) for transmission to the receiver's link layer. A POSITA would have understood that Shin's NULL primitive (also called a "NULL symbol") discloses an *idle character* because Shin's NULL symbols are inserted from symbol streams for clock synchronization using plesiosynchronous communication (Ex.1004, [0163]), like the '307 Patent's idle characters. Ex.1001, 49:4-7 ("[A] special idle character" can be inserted "between valid code words (each indicative of contents of a cell) for plesiosynchronous communication."); *id.* ("A slow receiver can drop (ignore) the idle characters to catch up to a faster transmitter."); Ex.1003, ¶¶114-115. A POSITA would have understood that Shin's NULL primitive is a form of *flow control* because it "account[s] for variations in clock frequency between the transmitting and receiving communications nodes." Ex.1004 at [0163]; Ex.1003, ¶115.

Shin further discloses that *the physical and link layer circuitry* of the receiver is configured to perform this simplex link flow control, because the NULL

primitives are inserted by the physical layer. Ex.1004 at [0169] (“NULL insertion and deletion [occurs] at the physical layer.”). This is illustrated in annotated Figure 9C below, which shows the *physical layer* of a receiver (RX) (**green**) adding or dropping NULL primitives in a sequence of symbols, illustrated as “DEC_NULL,” (**orange**). Ex.1003, ¶116.

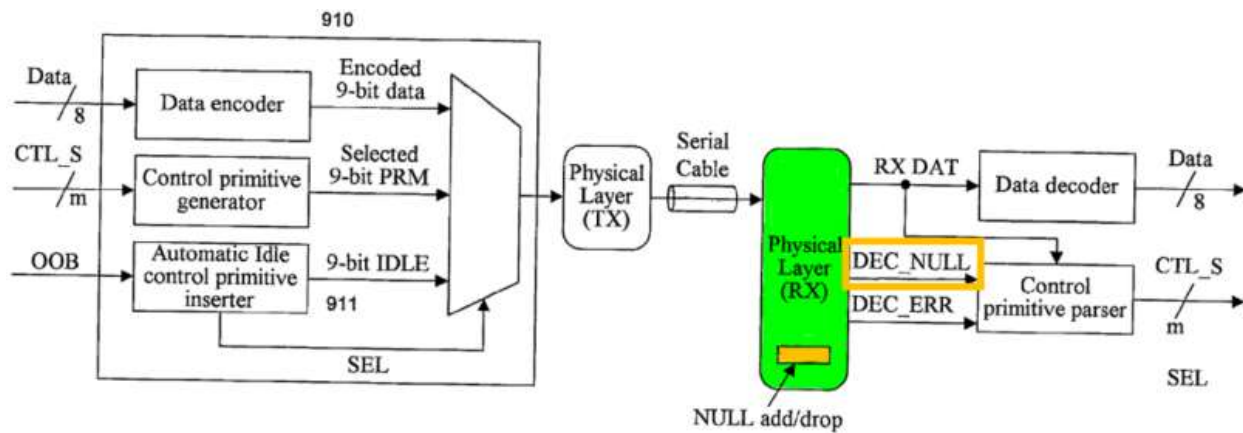


Fig. 9C

A POSITA would have understood that Shin inserts the NULL primitive in one direction (i.e., simplex link) for flow control because the receiver inserts the NULL primitive to align the receiver’s clock with the frequency of the transmitter, thus synchronizing the receiver with the transmitter. Ex.1004 at [0163]; Ex.1003, ¶117. Figure 9C above also shows a one-directional insertion of NULL primitives at the receiver’s physical layer, as illustrated by the one-directional arrow from the physical layer to the control primitive parser annotated in **orange**, thus illustrating

the use of the NULL primitives as a form of simplex link flow control. Ex.1003, ¶117.

Shin discloses and renders obvious insertion of NULL primitives (*idle characters*) *between subsequences of the code word sequence, where each of the subsequences is indicative of a different cell of the input words.* Ex.1003, ¶118.

Shin teaches that if its “sync and null detector” detects a synchronization primitive, it signals a “symbol pointer tracker” to “insert a NULL symbol into the stream of symbols.” Ex.1004, [0170], [0172]. Further, Shin discloses inserting NULL symbols between code words of transmitted packets for the following reasons. First, an IDLE primitive (which “may be a synchronization primitive”) can be inserted between the code words of the transmitted packets, since Shin’s packets are transmitted in a symbol stream of sequences of symbols (*code word sequence*), Ex.1004 at [0097] Fig. 9A; *id.*, [0079]. This is shown in Figure 9A, where an IDLE symbol (**green**) is being transmitted between packet 902 and the packet labeled as 907 (**orange**).

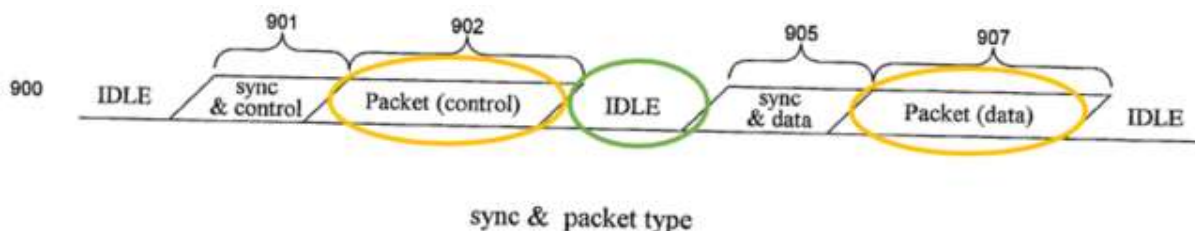


Fig. 9A

And, a POSITA would have understood that, since Shin’s sync and null detector signals for the NULL symbol to be inserted within the symbol stream when it receives a synchronization primitive, the NULL symbol (*idle character*) would likewise be inserted between different packets (*cells*), like the IDLE primitives. Ex.1003, ¶119. Therefore, the NULL symbols are inserted between subsequences of symbols indicative of different packets (*cells*) encoded from each packet’s respective codes (*input words*). Ex.1003, ¶119.

Additionally, Shin also discloses that “when the receiver detects a synchronization symbol, it initializes a start-of-symbol pointer in the buffer.” Ex.1004, [0073]. A “tracker ... inserts a NULL symbol at the start-of-symbol pointer. *Id.*, [0178]. A POSITA would have also understood that, since the NULL symbol is inserted at the start-of-symbol pointer upon detecting the synchronization symbol, the NULL symbol is inserted between packets, since the synchronization symbol (IDLE symbol) is transmitted between packets. Ex.1003, ¶120.

6. Claim 53

[53(pre)] A receiver for receiving encoded data that have propagated over a serial link, the encoded data having been generated in accordance with a line code, said receiver including:

To the extent the preamble is limiting, Shin discloses this limitation for the reasons discussed at limitation [1(d)]. Ex.1003, ¶121.

[53(a)] physical and link layer circuitry, coupled and configured to receive and decode the encoded data

Shin discloses *physical and link layer circuitry* that are *coupled* for the reasons discussed at limitation [1(b)]. Shin discloses *physical and link layer circuitry configured to received and decode the encoded data* for the reasons discussed at limitation [1(e)]. Ex.1003, ¶¶122-124.

[53(b)] to recover application data and control bits

Shin discloses this limitation for the reasons discussed at limitation [1(f)]. Ex.1003, ¶¶125-126.

[53(c)] to perform at least one link level function in response to at least one of the control bits; and

Shin discloses this limitation for the reasons discussed at limitation [1(g)]. Ex.1003, ¶¶127-128.

[53(d)] higher level circuitry coupled to the physical and link layer circuitry, wherein the physical and link layer circuitry is coupled and configured to assert at least one control signal indicative of at least another one of the control bits to the higher level circuitry,

Shin discloses and renders obvious this limitation for the reasons discussed at limitations [1(b)] and [1(h)]. Ex.1003, ¶¶129-132 (describing that a POSITA would have understood the physical and link layer circuitry would be coupled to the higher-level circuitry).

[53(e)] and the higher level circuitry is configured to perform at least one higher level function in response to the at least one control signal.

Shin discloses this limitation for the reasons discussed at limitation [1(i)].
Ex.1003, ¶¶133-134.

D. Ground 3: Claims 1, 2, 6, 7, 23, 25, 31, 53, 68, 70, 71, 72, 74, and 75 are Obvious over Yusairi in view of Shin

1. Summary of Yusairi

Yusairi describes using an IPv4-over-1394 protocol stack for communication between devices. Ex.1005, 51, Sec. 1. Each device includes a physical layer and a link layer, along with two higher-level network and transport layers. *Id.*, 53-54, Sec. 5, 5.1. The link layer and physical layer, which are part of the IEEE1394 interface (also known as “FireWire”), perform transmit and receive operations between devices. *Id.*, 51, Sec. 1; *id.*, 52-53, Sec. 3.2, Fig. 1.

When packets are generated by stream_trans⁴ at the link layer of the transmitter, a link header is attached to the data. *Id.*, 54, Sec. 5.2. The link header includes a “transaction code.” *Id.* When received at the transmitter by stream_rec, the link layer performs a header check process based on the value of the transaction code. *Id.* During the header check process, stream_rec checks the type field. *Id.*, 55, Sec. 5.2. The link layer then signals the network layer based on the value of the type field. *Id.* When ip_rec of the network layer detects the ready signal from stream_rec

⁴ Yusairi uses bold font when referring to various module names. When discussing or quoting from Yusairi, this Petition removes the bold font emphasis.

of the link layer, it performs several functions, including reading fifo, checking a destination address of the packet, and calculating an IP checksum, each of which are network layer functions. *Id.*, 55, Sec. 5.3.

2. Summary of Shin

See Sec. IX.C.1, *supra*.

3. Claim 1

[1(pre)] A communication system, including:

Yusairi discloses this limitation. Yusairi discloses *a communication system* between two boards, SOPC Board A transmitting to SOPC Board B, illustrated in Figure 6, below. Ex.1005 at 56, Sec. 7.1.1-7.1.2, Fig. 6; Ex.1003, ¶¶138-139.

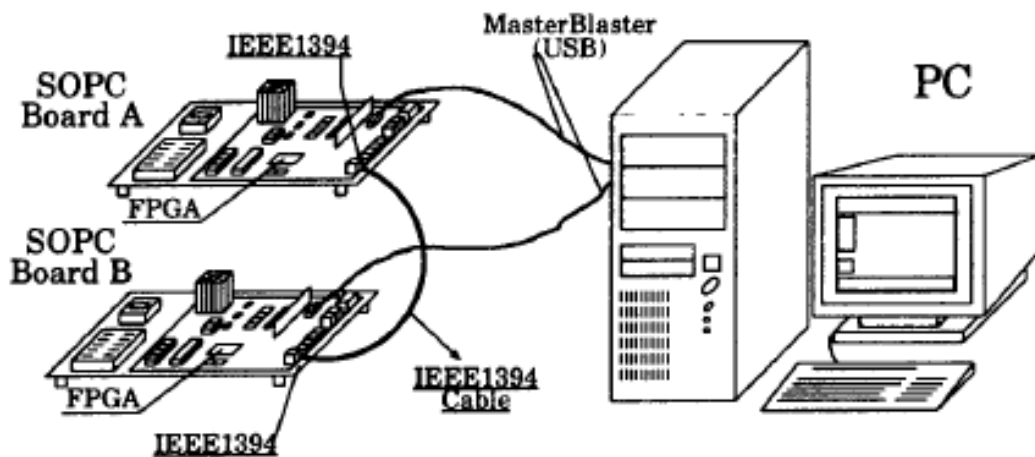


Fig. 6 System used for experiment (1).

[1(a)] a first endpoint;

Yusairi discloses this limitation. Figure 6 below is annotated to show SOPC Board A, which acts as *a first endpoint* because “SOPC board A generates data ...

[that is] transmitted to the SOPC board B.” Ex.1005, 56, Sec. 7.1.2; Ex.1003, ¶¶141-142.

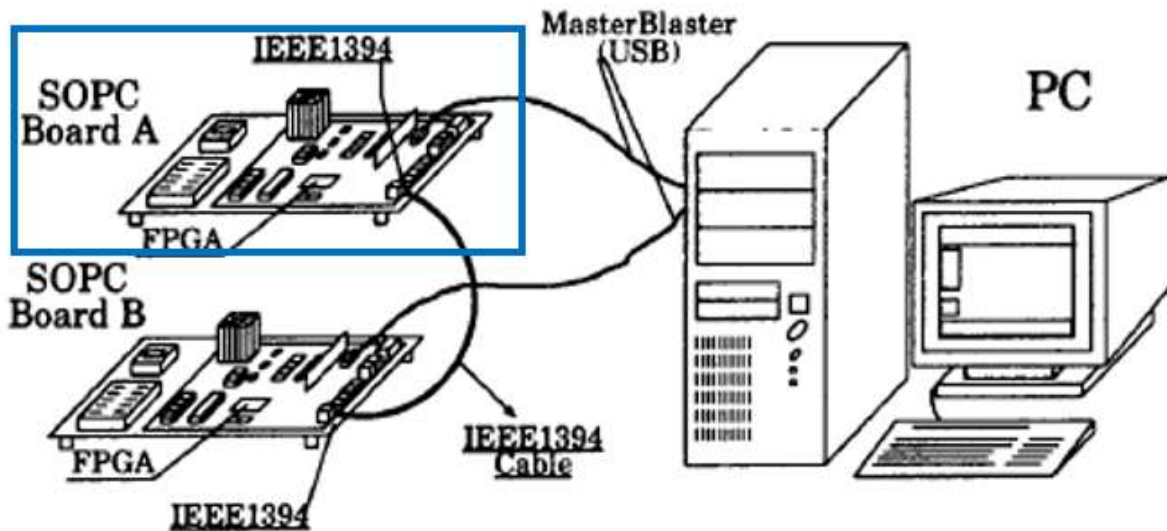


Fig. 6 System used for experiment (1).

[1(b)] a second endpoint, including physical and link layer circuitry and higher level circuitry coupled to the physical and link layer circuitry; and

Yusairi discloses this limitation. Figure 6 below is annotated to show SOPC Board B, which acts as *a second endpoint* because it receives packets from SOPC Board A. Ex.1005, 56, Sec. 7.1.2; Ex.1003, ¶¶143-144.

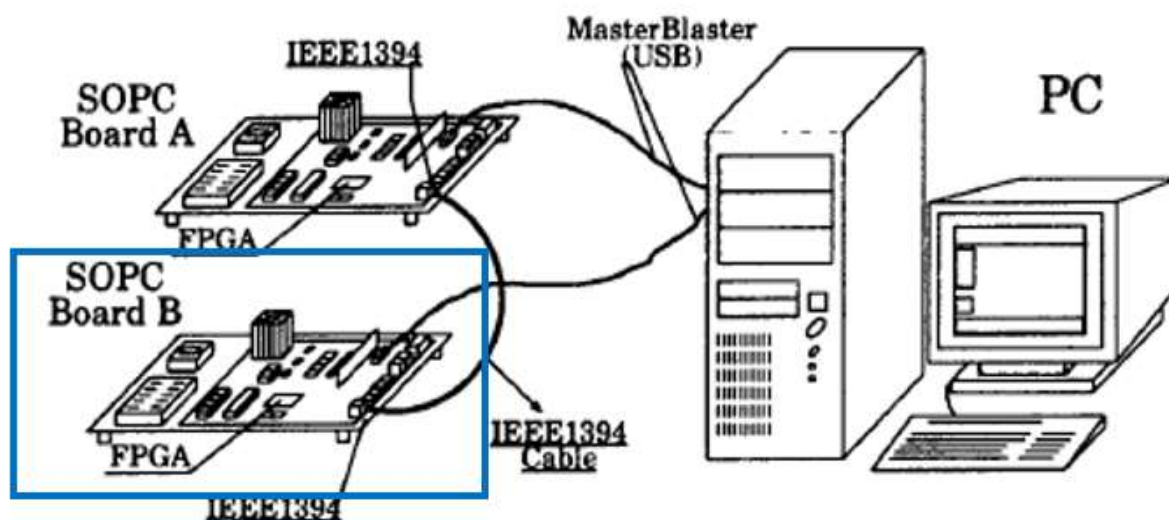


Fig. 6 System used for experiment (1).

The *second endpoint*, SOPC Board B, includes *physical and link layer circuitry* because it includes an “LLC-PHY interface.” *Id.*, 54, Sec. 5.1; *see also id.*, 53, Sec. 3.2 (“The IEEE1394 interface consists of link and physical layers which are usually called LLC and PHY, respectively.”); Ex.1003, ¶145.

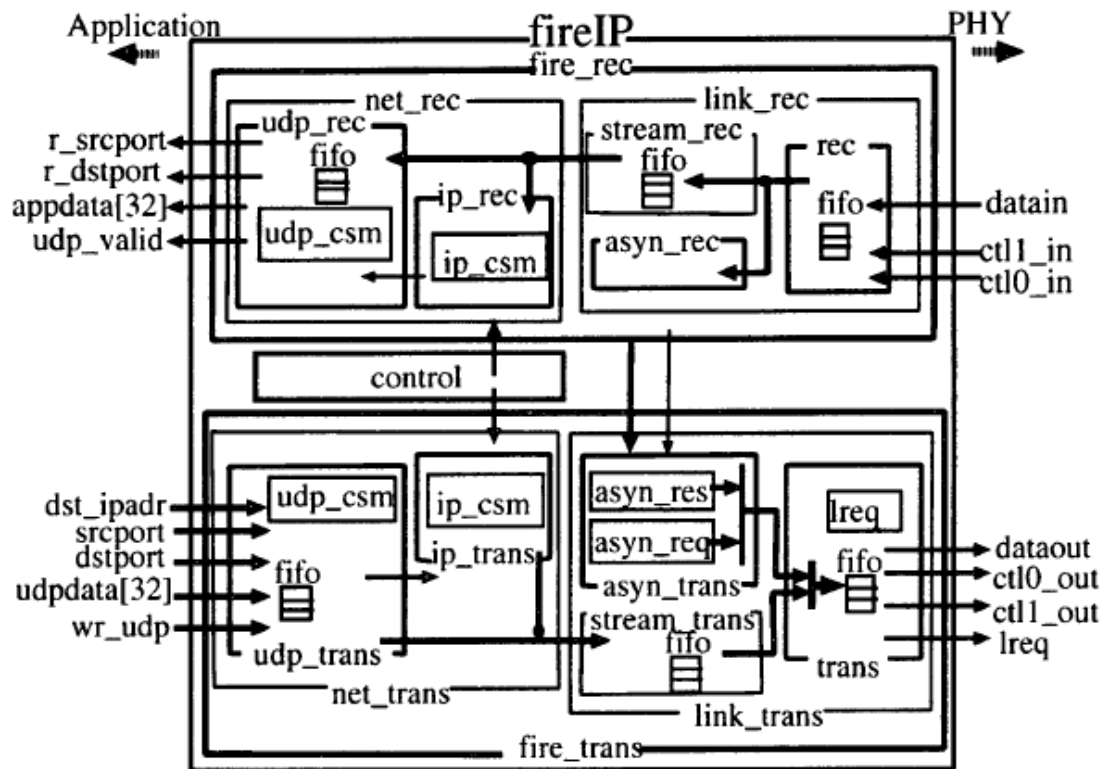


Fig. 5 Block diagram of fireIP module.

Further, the FireIP module includes circuitry corresponding to modules controlling the link layer and physical layer interface (respectively, *link layer circuitry* and *physical layer circuitry*), including “fifo,” “crc,” “lreq,” “trans,” and “rec.” *Id.*, 54, Sec. 5.1; Ex.1003, ¶146.

The FireIP module also shows *higher level circuitry coupled to the physical and link layer circuitry*. The FireIP modules includes a “network layer” comprised of circuitry (*higher level circuitry*) that corresponds to functions performed at the network layer, such as “ip_trans and ip_rec for transmitting and receiving.” *Id.*, 54, Sec. 5.3; Ex.1003, ¶147.

A POSITA would have understood that Yusairi’s link and physical layers are coupled to the higher-level network and transport layers because components of the PHY-LLC interface communicate packets to the higher-level network and transport layers, thus disclosing *higher level circuitry coupled to the physical and link layer circuitry*. *Id.*, 55, Sec 5.2 (At the link layer, “the packet is accepted and handed to the ip_rec” module of the network layer); Ex.1003, ¶148 (describing how a POSITA would have understood that Yusairi’s higher level circuitry is coupled to the lower level physical and link layer circuitry in view of Figure 4).

[1(c)] a serial link between the first endpoint and the second endpoint,

Yusairi discloses this limitation. An IEEE1394 cable (*the serial link*) is used to connect the SOPC Board A (*the first endpoint*) and the SOPC Board B (*the second endpoint*) as annotated below in Figure 6. Ex.1005 at 56, Sec. 7.1.1, FIG. 6. A POSITA would have understood an IEEE1394 cable to be a *serial link* because since it transmits data from SOPC Board A to SOPC Board B. Ex.1003, ¶¶149-150 (citing Ex.1008 at 46, which discusses an IEEE1934 bus that transmits data between nodes).

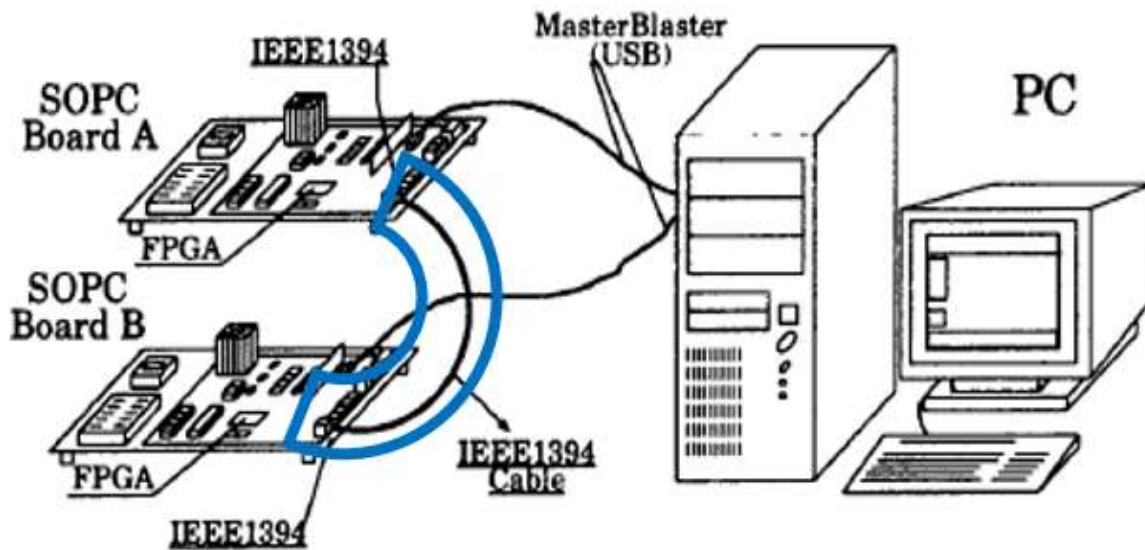


Fig. 6 System used for experiment (1).

[1(d)] wherein the first endpoint is configured to generate encoded data in accordance with a line code, and to transmit the encoded data over the link to the second endpoint, and

Yusairi discloses this limitation. Yusairi discloses that at SOPC board A (*the first endpoint*), packets are “encapsulated and transmitted to the SOPC board B” (*generate encoded data ... and to transmit the encoded data over the link to the second endpoint*). Ex.1005, 56, Sec. 7.1.2. At SOPC board B, “[r]eceived packets [are] decapsulated.” *Id.* A POSITA, therefore, would have understood that Yusairi is encoding data by encapsulating it at SOPC board A. Ex.1003, ¶¶151-152.

Yusairi also discloses using IEEE1394, which a POSITA would have understood encodes data using an NRZ line code (*in accordance with a line code*). Ex.1005, 56, Sec. 7.1.1 (stating that the Boards are equipped with an IEEE1394 PHY

chip); Ex.1003, ¶153 (describing how the IEEE1394 standard uses NRZ line code to communicate over a bus) (citing Ex.1008, 15, 81-82, 95).

[1(e)] wherein the physical and link layer circuitry of the second endpoint is coupled to the link and configured to decode the encoded data

Yusairi discloses this limitation. As illustrated in Figure 6, Yusairi's SOPC Board B (*the second endpoint*) is coupled to the IEEE1394 cable (*the link*) over which it receives the packets from SOPC Board A. Ex.1005, 56, Sec. 7.1.1, Fig. 6; Ex.1003, ¶¶154-155.

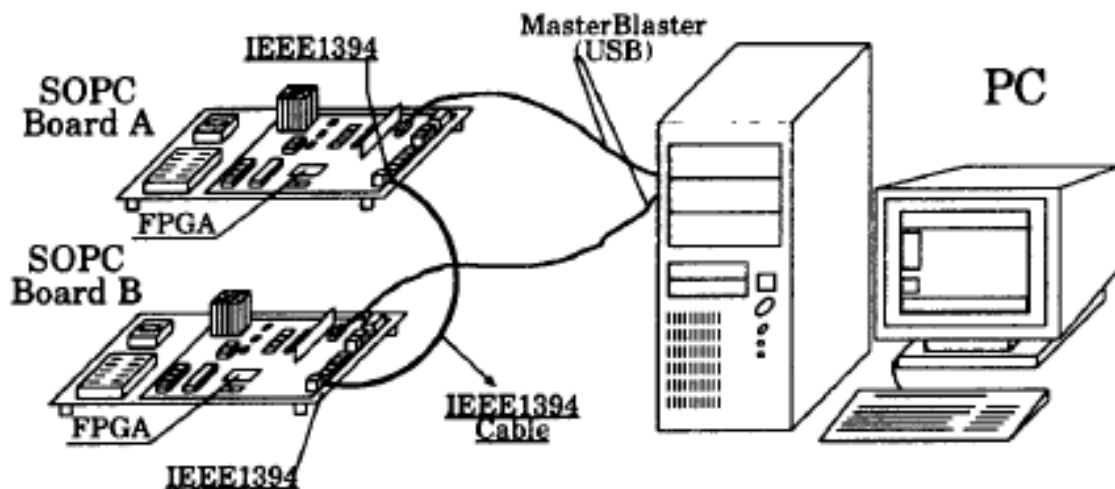


Fig. 6 System used for experiment (1).

Further, as illustrated below in Figure 4, the physical layer of the LLC-PHY interface at SOPC Board B receives the packets from SOPC Board A. Ex.1005, 53, FIG. 4; Ex.1003, ¶156.

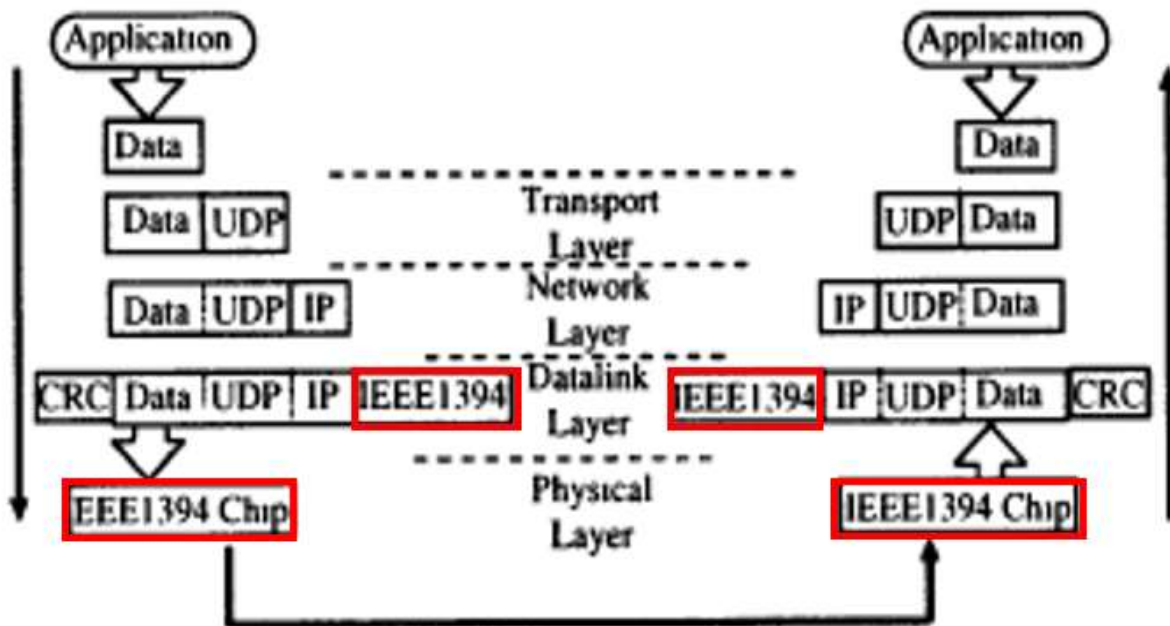


Fig. 4 UDP/IP protocol suite.

Thus, a POSITA would have understood that the physical and link layer circuitry of the SOPC Board B (*the second endpoint*) is coupled to the IEEE1394 cable (*the link*) to receive the packets from SOPC Board A at the physical layer. Ex.1003, ¶156.

Yusairi further discloses that the physical and link layer circuitry of the second endpoint *is configured to decode* the encoded data packets. At SOPC Board A, packets are “encapsulated and transmitted to the SOPC Board B.” *Id.*, 56, Sec. 7.1.2. At SOPC Board B, “[r]eceived packets [are] decapsulated.” *Id.* A POSITA would have also understood that the encoded data is decoded by *the physical and link layer circuitry of SOPC Board B (the second endpoint)*. Ex.1003, ¶157. That is because the “PHY [physical layer] hands a received packet to the module rec” and the “rec

module then puts the packet into a fifo.” Ex.1005, 54, Sec. 5.1. At the link layer, “stream_rec takes the data from the fifo and checks the link header” to “start the header check process.” *Id.*, 54, Sec. 5.2. As such, a POSITA would have understood that encoded data is being decoded by the *physical and link layer circuitry* since it is being received by the physical layer and used at least by the link layer. Ex.1003, ¶159.

Further, a POSITA would have understood that, because Yusairi uses the IEEE1394 for the physical and link layer circuitry (annotated in **red** above) of the SOPC Board B, the encoded data would be decoded at the physical layer. Ex.1003, ¶158 (describing how the IEEE1394 standard calls for decoding received data at the physical layer) (citing Ex.1008, 132).

[1(f)] to recover application data and control bits,

Yusairi discloses this limitation. At SOPC Board A (the first endpoint), the stream_trans module “generates a link header, [and] attaches it to IP data.” Ex.1005, 54, Sec. 5.2. Within the link header, there is “a transaction code” and “a type field.” *Id.*, 54-55, Sec. 5.2. The link header and IP data packet are “transmitted to the SOPC Board B [*the second endpoint*]” *Id.*, 56, Sec. 7.1.2; Ex.1003, ¶¶160-161.

A POSITA would have understood the “transaction code” and the “type field” each include *control bits* because, consistent with the ’307 Patent, each causes performance of a function at one of the layers, such as the link layer and the network

layer. Ex.1001, 7:55-60 (discussing control bits “providing transport layer functionality”); *id.*, 7:60-65 (discussing control bits providing link layer functionality); Ex.1003, ¶162. For instance, in Yusairi, at the link layer, “If the transaction code is equal to 0xA ... stream_rec will start the header check process.” Ex.1005, 54, Sec. 5.2. Then, if the “type field equals 0x0800 (IPv4) the packet is accepted and handed to the ip_rec [of the network layer]” where ip_rec “checks the destination address,” “accept[s] the packet and calculates the IP checksum.” *Id.*, 55, Sec. 5.2-5.3. Thus, a POSITA would have understood that the transaction code includes *control bits* because they cause the link level function of starting a header check process, and the type field includes *control bits* because it causes higher-level functions at the network layer, such as checking the destination address, accepting the packet, and calculating the IP checksum. Ex.1003, ¶162.

As noted above at limitation [1(e)], the physical and link layers of SOPC Board B decode the received packet, which would “recover” the control bits. A POSITA would have understood that the *control bits* of the transaction code and type field are “recovered” because the link layer uses them to determine the value of the transaction code “to start the header check process” and the value of the type field to assert a “ready signal” to the network layer. Ex.1005, Sec. 5.2-5.3; Ex.1003, ¶163.

Further, Yusairi also discloses recovering *application data*. As noted, the header is attached to “IP data.” Ex.1005, 54, Sec. 5.2. A POSITA would have

understood that the IP data (*the application data*) is recover[ed] because the IP data is included in the packet, and the packet is decoded so that it can be “accepted and handed to the ip_rec [module of the network layer].” *Id.*, 55, Sec. 5.2; Ex.1003, ¶164 (discussing that a POSITA would have understood Yusairi’s IP data to be application data and that the application data is recovered by the physical and link layers so that it can be accepted and handed to the network layer).

[1(g)] to perform at least one link level function in response to at least one of the control bits, and

Yusairi discloses this limitation. For the reasons discussed at limitation [1(f)], Yusairi discloses “a transaction code” and “a field type” in a packet header that each comprise *control bits*. In response to at least one of the control bits of the transaction code, “stream_rec [a link level module] will start the header check process [*a link level function*].” Ex.1005, 54, Sec 5.2 The header check process (*the link level function*) occurs *in response to at least one of the control bits* because the header check process (*link level function*) starts when “the transaction code is equal to 0xA.” *Id.*, 54, Sec 5.2; Ex.1003, ¶¶165-166.

[1(h)] to assert at least one control signal indicative of at least another one of the control bits to the higher level circuitry, and

Yusairi discloses this limitation. As discussed in above with respect to limitation [1(g)], Yusairi discloses that a link level function (the header check

process) is performed in response to at least one of the control bits from the transaction code. Ex.1005, 54, Sec 5.2; Ex.1003, ¶167.

Yusairi further discloses *assert[ing] at least one control signal indicative of at least another one of the control bits to the higher level circuitry*. Yusairi discloses that, at the link layer, “the packet is accepted and handed to the ip_rec [module of the network layer]” when “the type field [is] 0x0800.” Ex.1005, 55, Sec. 5.2-5.3. Accordingly, the ip_rec “module [of the network layer] reads its fifo when it detects [a] ready signal [*the control signal*] asserted by stream_rec [a link layer module].” *Id.*, 55, Sec. 5.3. Thus, Yusairi asserts *at least one control signal* (the ready signal) *to the higher level circuitry* (the network layer). Ex.1003, ¶168.

Further, the ready signal (*the control signal*) is *indicative of* the bits of the type field (*control bits*) because it occurs in response to “the type field [being] equal[] to 0x800,” thus indicating the presence and value of the type field control bits. Ex.1005, 55, Sec. 5.2; Ex.1003, ¶169.

Additionally, a POSITA would have understood that the transaction code bit(s) are different from the type field bit(s) because the type field value is determined during the header check process that starts in response to the transaction code bit value. Ex.1003, ¶170. Thus, the transaction code includes at least one control bit, while separately, the type field includes *at least another one of the control bits*. *Id.*

[1(i)] wherein the higher level circuitry is configured to perform at least one higher level function in response to the at least one control signal.

Yusairi discloses and renders obvious this limitation. As discussed with respect to limitation [1(h)], Yusairi's network layer includes *higher level circuitry*, and it receives a ready signal (*the control signal*) asserted from the link layer. Ex.1003, ¶171.

Additionally, Yusairi discloses that the network layer (*higher level circuitry*) is configured to perform several functions, including “reading fifo,” “check[] the destination address,” and “calculate[] the IP checksum.” Ex.1005, 55, Sec 5.3. A POSITA would have recognized these are each a *higher level function* since they are performed by the network layer (*the higher level circuitry*). Ex.1003, ¶172. These network-layer functions are performed *in response to the at least one control signal* because they occur “when [ip_rec] detects the ready signal.” Ex.1005, 55, Sec 5.3.

4. Claim 2

[2] The system of claim 1, wherein the line code specifies a block code and special characters, the first endpoint is configured to encode M-bit input words in accordance with the block code to generate a code word sequence of N-bit code words, where $N > M$, the code word sequence is indicative of at least one cell of the input words, and the encoded data are a sequence of at least one of the special characters and the code words of the code word sequence.

Yusairi in view of Shin renders obvious this limitation. Shin discloses this limitation for the reasons discussed with respect to limitation [2] of Grounds 1 and 2, which discusses Shin's use of block code (such as 8B/9B), and transmitting block

encoded in-band symbols and out-of-band special characters via the line code. A POSITA would have understood that using Shin's block code to encode Yusairi's packets (having the application data and control bits) would generate encoded in-band symbols, which is the block code from the packets that is included with the special characters, like IDLE primitives, when transmitted in line code. Ex.1004, [0086] ("A packet that is transmitted on a serial communications link typically comprises in-band symbols."); *id.*, [0131] (discussing 8B/9B encoding generates 9-bit symbols from 8-bit codes); *id.*, [0086] (further discussing that out-of-band symbols, like IDLE characters, can be inserted into in-band symbols and transmitted between communications links); Ex.1003, ¶¶175-176.

A POSITA would have been motivated to use Shin's block encoding methods to encode Yusairi's packets because Shin's block encoding methods, encoding M-bit codes to N-bit symbols, where $N > M$, would have provided Yusairi with known communication protocols having a neutral DC component to help ensure signal is accurately received and decoded by synchronizing the SOPC Board B receiver with the SOPC Board A transmitter. Ex.1003, ¶177 (citing Ex.1010, 55) (discussing known prior art systems that used $N > M$ block encoding methods that provided DC balance). Further, a POSITA would have been motivated to modify Yusairi with Shin's encoding methods because doing so would have provided for flow control at the lower physical and link layers, as opposed to relying on higher-level application

functions to provide flow control. Ex.1005, 53, Sec. 4.1 (“For more sophisticated reliabilities or flow control necessary for each application, we rely on functions to be supplied by the application itself.”); Ex.1003, ¶177 (citing Ex.1009, 129) (discussing link layer functions of prior art systems that performed block encoding and flow control).

5. Claim 6

[6] The system of claim 2, wherein said at least one of the special characters is an idle character.

Yusairi in view of Shin renders obvious this limitation for the reasons discussed with respect to limitation [2].

A POSITA would have been motivated use Shin’s encoding methods with Yusairi for the reasons discussed with limitation [2]. Additionally, Shin discloses that IDLE primitives (*idle characters*) would help synchronize transmitters and receivers, and as such, a POSITA would have been further motivated to use Shin’s IDLE primitives (*idle characters*) with Yusairi’s system to enhance synchronization between SOPC Board A and SOPC Board B. Ex.1004, [0096] (“An IDLE primitive may be a synchronization primitive.”); *id.*, [0096] (“When a communications node receives a synchronization primitive, it knows that the primitive is correctly aligned on a symbol boundary.”); Ex.1003, ¶¶178-179.

6. Claim 7

[7] The system of claim 6, wherein the physical and link layer circuitry is configured to perform simplex link flow control by inserting idle characters between subsequences of the code word sequence, where each of the subsequences is indicative of a different cell of the input words.

Yusairi in view of Shin renders obvious this limitation. Shin discloses this limitation as discussed at limitation [7] of Grounds 1 and 2. Further, a POSITA would have been motivated to modify Yusairi with a communication architecture that uses simplex link flow control for the reasons also stated at limitations [2] and [6], e.g., to provide Yusairi with lower physical and link layer flow control capabilities rather than relying on higher application layers for flow control. Ex.1003, ¶180.

7. Claim 23

[23(pre)] A communication system, including:

Yusairi discloses this limitation for the reasons set forth in [1(pre)]. Ex.1003, ¶181.

[23(a)] a first endpoint;

Yusairi discloses this limitation for the reasons set forth in [1(a)]. Ex.1003, ¶182.

[23(b)] a second endpoint; and

Yusairi discloses this limitation for the reasons set forth in [1(b)]. Ex.1003, ¶183.

[23(c)] a serial link between the first endpoint and the second endpoint,

Yusairi discloses this limitation for the reasons set forth in [1(c)].
Ex.1003, ¶184.

[23(d)] wherein the first endpoint is configured to encode M-bit input words in accordance with a block code to generate a code word sequence of N-bit code words, where $N > M$,

Yusairi in view of Shin renders obvious this limitation for the reasons set forth in [2]. A POSITA would have been motivated to modify Yusairi with Shin's encoding methods for the reasons discussed at limitation [2]. Ex.1003, ¶¶185-186.

[23(e)] the code word sequence is indicative of at least one cell of the input words, the input words of each said cell are indicative of application data and control bits,

Yusairi in view of Shin renders obvious this limitation. First, Shin discloses *the code word sequence is indicative of at least one cell of the input words* for the reasons stated at limitation [2]. Ex.1003, ¶¶187-188.

Further, Yusairi in view of Shin renders obvious *the input words of each said cell are indicative of application data and control bits*. As discussed at [1(f)], Yusairi discloses generating a packet having *application data* (IP data) and *control bits* (transaction code bits and field type bits) at SOPC Board A. Ex.1005, 54, Sec. 5.2. As discussed at limitation [2] under Ground 1 and 2, Shin discloses that a packet (*cell*) can be formed into codes (*input words*) that are encoded by block encoding into symbols (*code words*). A POSITA would have understood that when Yusairi's

packet having application data (IP data) and control bits (transaction code and type field) are encoded using Shin's block code, the application data and control bits from the packet (*cell*) would have been placed into codes (*input words*) before being encoded into symbols (code words). Ex.1003, ¶189. Thus, by applying Shin's block encoding to Yusairi's packets, the codes (*input words*) of each packet (*cell*) would be indicative of application data and control bits. Ex.1003, ¶189.

A POSITA would have been motivated to combine Shin's teaching of using a *code word sequence [] indicative of at least one cell of the input words* with Yusairi's teaching of *the input words of [a] cell [being] indicative of application data and control bits* because doing so would have allowed Yusairi's packet having application data and control bits to be encoded and transmitted as part of a sequence of symbols (*code word sequence*) using known block encoding methods, such as Shin's 8B/9B block encoding. Ex.1004, [0131]; Ex.1003, ¶190. A POSITA would have been further motivated to encode Yusairi's packets with Shin's block encoding methods for the reasons stated at limitations [2] and [6]. Ex.1003, ¶190.

[23(f)] the control bits have multiple levels of communication protocol functionality, and

Yusairi discloses this limitation. As discussed at limitation [1(f)], Yusairi discloses a transaction code and a field type having *control bits*. As noted in [1(g)], the transaction code control bits are functional at the link level because, "If the

transaction code is equal to 0xA [the link layer] will start the header check process [a communication protocol functionality at the link layer].” Further, as described in [1(i)], the transport layer performs any one of several functions—“checks the destination address,” “accept[s] the packet and calculates the checksum”—responsive to a ready signal asserted when “the type field [is] equal[] to 0x0800.” See also, Ex.1005, 55, Sec. 5.2, 5.3. A POSITA would have understood that Shin teaches this limitation because a function is performed at the link layer and another function performed at the transport layer (i.e., multiple levels) based on values of the transaction code and the type field. Ex.1003, ¶¶191-192.

[23(g)] the first endpoint is configured to transmit encoded data including the code words of the code word sequence over the link to the second endpoint.

Yusairi in view of Shin renders obvious this limitation. As discussed in [1(d)], Yusairi discloses that *the first endpoint is configured to transmit encoded data over the link to the second endpoint*. Ex.1003, ¶¶ 193-194.

Further, Shin discloses that *encoded data includes the code words of the code word sequence* for the reasons discussed in limitation [2] of Grounds 1 and 2. A POSITA would have been motivated to modify Yusairi with Shin’s encoding methods to render obvious transmit[ing] encoded data that includes symbols (*code words*) of a sequence of symbols (*code words sequence*) for the reasons discussed for limitations [2] and [6]. Ex.1003, ¶195.

8. Claim 25

[25] The system of claim 23, wherein the block code is specified by a line code that also specifies special characters, the encoded data are a sequence of at least one of the special characters and the code words of the code word sequence, and none of the special characters of the encoded data matches a bit sequence of the code word sequence.

Yusairi in view of Shin renders obvious this limitation.

As discussed at limitation [2], Shin discloses *wherein the block code is specified by a line code that also specifies special characters, the encoded data are a sequence of at least one of the special characters and the code words of the code word sequence.* Ex.1003, ¶¶196-197.

Further, Shin discloses or renders obvious that *none of the special characters of the encoded data matches a bit sequence of the code word sequence.* Shin's transmitted packets comprise in-band symbols (*the code word sequence*), and the in-band symbols include bits (*bit sequence of the code word sequence*). *Id.*, [0086]. Shin also teaches that control characters (*the special characters of the encoded data*) “have a unique number of transitions that is not found in any ... in-band symbol [*code word sequence*].” *Id.*, [0101]. A POSITA would have understood that Shin's control characters with transitions that are not found in bits of any in-band symbols discloses that *none of the special characters of the encoded data matches a bit sequence of the code word sequence.* Ex.1003, ¶198.

A POSITA would have been motivated to use Shin’s block encoding methods, which includes out-of-band symbol that do not normally appear within a packet, with Yusairi for the reasons discussed for limitations [2] and [6], and because other communication systems at the time also used out-of-band symbols that did not normally appear within a packet. Ex.1003, ¶198-199 (citing Ex.1009, 142-43) (discussing prior known methods, such as SATA, that used special characters that did not match other conveyed data).

9. Claim 31

[31] The system of claim 23, wherein each of the first endpoint and the second endpoint is a transceiver, and the second endpoint is configured to generate additional encoded data in response to input words in accordance with the line code and to transmit the encoded data over the link to the first endpoint.

Yusairi discloses this limitation and Yusairi in view of Shin also renders obvious this limitation. Each of SOPC Board A (*the first endpoint*) and the SOPC Board B (*the second endpoint*) has the FireIP component that “consists of fire_trans and fire_rec for transmit and receive operations, respectively....” Ex. 1005, 53, Sec. 5; *see also Id.*, 55, Sec. 7.1 (“communications between two fireIP modules.”). Because each of the Boards can perform “transmit and receive operations,” a POSITA would have understood each of these endpoints is *a transceiver*. *Id.*, 53, Sec. 5; Ex.1003, ¶¶200-201.

In addition, Shin discloses *the second endpoint is configured to generate additional encoded data in response to input words in accordance with the line code and to transmit the encoded data over the link to the first endpoint*. Shin describes a communication node 3801 (*the first endpoint*) and another communication node 3811 (*the second endpoint*) that are each a *transceiver* because both have a transmitter component and a receiver component and can communicate using a bidirectional serial link. Ex.1004, [0164]; Ex.1003, ¶202. This is illustrated by Figure 38A below, annotating the transmitter and receiver (purple).

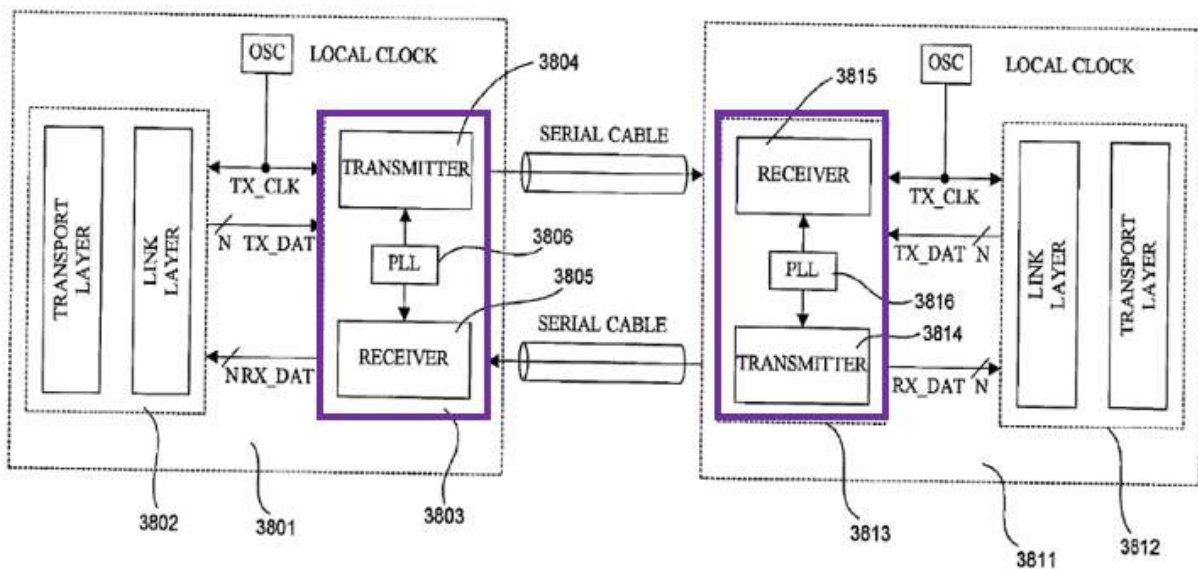


Fig. 38A

As illustrated in Figure 38A, encoded data is transmitted over a serial cable (*the link*) from communication node 3811 (*the second endpoint*) to communication node 3801 (*the first endpoint*). Ex.1004, Fig. 38A. Further, Shin discloses that nodes

can transmit N-bit block *encoded in response* to M-bit *input words*, such as 8B/9B block encoding. *Id.*, [0128], [0131], [0165]. As discussed at [1(d)], a POSITA would have understood that Shin's 8B/9B block code is *generated in accordance with line code*. Thus, because communication node 3811 (*the second endpoint*) is a communication node that can encode codes (*input words*) to symbols (*code words*) that are transmitted by a line code to communication node 3801 (*the first endpoint*), a POSITA would have understood that Shin discloses this limitation. Ex.1003, ¶203.

A POSITA would have been motivated to combine Shin's teaching of generating additional encoded data in response to input words in accordance with the line code at the second endpoint and transmitting it to the first endpoint because doing so would provide full-duplex communication, which was standard for IEEE1394, used for the link and physical layer circuitry, and the IEEE1394 cable communications link. Ex.1003, ¶204 (citing Ex.1008, 15, 34) (discussing the use of full-duplex by IEEE1394).

10. Claim 53

[53(pre)] A receiver for receiving encoded data that have propagated over a serial link, the encoded data having been generated in accordance with a line code, said receiver including:

To the extent the preamble is limiting, Yusairi in view of Shin renders obvious this limitation for the reasons set forth in [1(b)] and [1(d)]. Ex.1003, ¶205.

[53(a)] physical and link layer circuitry, coupled and configured to receive and decode the encoded data

Yusairi discloses this limitation and Yusairi in view of Shin renders this limitation obvious for the reasons discussed in [1(e)]. Ex.1003, ¶206.

[53(b)] to recover application data and control bits

Yusairi discloses this limitation for the reasons discussed in [1(f)]. Ex.1003, ¶207.

[53(c)] to perform at least one link level function in response to at least one of the control bits; and

Yusairi discloses this limitation for the reasons discussed in [1(g)]. Ex.1003, ¶208.

[53(d)] higher level circuitry coupled to the physical and link layer circuitry, wherein the physical and link layer circuitry is coupled and configured to assert at least one control signal indicative of at least another one of the control bits to the higher level circuitry,

Yusairi discloses this limitation for the reasons discussed in [1(b)] and [1(h)]. Ex.1003, ¶209.

[53(e)] and the higher level circuitry is configured to perform at least one higher level function in response to the at least one control signal.

Yusairi discloses this limitation for the reasons discussed in [1(i)]. Ex.1003, ¶210.

11. Claim 68

[68(pre)] A transmitter including physical and link layer circuitry, wherein the physical and link layer circuitry include:

To the extent the preamble is limiting, Yusairi discloses this limitation. Yusairi's SOPC Board A is *a transmitter* because it "generates data ... [that is] transmitted to the SOPC board B." Ex.1005, 56, Sec. 7.1.2. Further, SOPC Board A (*the transmitter*) includes a "Datalink"/"Link" layer using "IEEE1394" and a "physical layer" using an "IEEE1394 Chip" (*Id.*, 53, Fig. 4 and Table 1), and may be implemented on a "fireIP" module containing a "LLC-PHY interface" (*link layer and physical layer circuitry*). *Id.*, 53, Sec. 5; 54, Sec. 5.1; *see also id.*, 55, Sec. 7.1 ("two fireIP modules"). Ex.1003, ¶¶211-212.

[68(a)] packetizing circuitry configured to generate M-bit input words,

Yusairi in view of Shin renders this limitation obvious. Yusairi discloses the SOPC Board A (*the transmitter including physical and link layer circuitry*) comprises *packetizing circuitry*, as it includes the fireIP module, which includes a link layer and physical layer, as discussed for [68(pre)], and the link layer encapsulates data into packets and passes them on for transmission. Ex.1005, 54, Secs. 5, 5.2; 56, Sec. 7.1.2; 53, Secs. 4 4.3; Ex.1003, ¶214. Specifically "the module link_trans for the link layer" does so by "generat[ing] a link header, attach[ing] it to IP data, and put[ting] the data into a fifo of trans module." Ex.1005, 54, Secs. 5, 5.2. The header includes "encapsulation headers." *Id.*, 54, Sec. 5.2. Then, "[w]hen the packet is ready to be transmitted, a signal is asserted to inform [the] trans module that IP data has been put into the fifo and [is] ready to transmit." *Id.*, 54, Sec. 5.2.

Then, trans “starts transmitting link packets ... to PHY [physical layer]” and “request[s] PHY that the data be transmitted.” *Id.*, 54, Sec. 5.1. The function of link_trans combining headers and IP data into packets aligns with the ’307 patent’s description of its “*packetizing circuit*,” which “group[s]” “data blocks (sometimes referred to as words) ... into cells,” where cells can be packets. Ex.1001, 14:8-10, 4:38-40. Yusairi, therefore, discloses *link layer circuitry* that *include[s] packetizing circuitry*. Ex.1003, ¶214.

Yusairi in view of Shin renders obvious the *packetizing circuitry* being *configured to generate M-bit input words*. As discussed for [2] in Grounds 1 and 2, Shin’s packet (*cell*) can be formed into codes (*M-bit input words*) that are encoded by block encoding into symbols (*code words*). *See supra*, sec. IX.C.3. Shin describes an example block encoding that uses “8b/9b encoding,” which encodes “8-bit codes [to] ... 9-bit symbols.” Ex.1004, [0131]. A POSITA would have understood that block encoding encodes M-bit codes (*input words*), such as the 8-bit codes, to N-bit symbols (*code words*), such as the 9-bit symbols. *Id.*; Ex.1003, ¶215.

A POSITA would have been motivated to use Shin’s block encoding methods, which *generate M-bit input words*, with Yusairi’s link-layer *packetizing circuitry* in generating packets for transmission, because it was known such block encoding methods help maintain DC balance between transmitters and receivers, such as Yusairi’s SOPC Boards, which is a benefit taught by Shin. Ex.1003, ¶216; Ex.1004,

[0131]-[0132]. A POSITA would have also been motivated to use Shin’s block encoding methods with Yusairi’s link-level packetizing circuitry because it was a well-known industry practice to perform packetization and block encoding at the link layer. Ex.1003, ¶216 (discussing an HDMI “Link Architecture” that performs packet encoding and serialization) (citing Ex.1010, 40, Fig. 5-1); *id.*, ¶217 (discussing 802.3 (Ethernet) standard that creates packets at the link level) (citing Ex.1011, 98, 896, 928, Figs. 35-1, 36.3). Further, Yusairi’s teachings regarding performing packetization in the link layer would have motivated a POSITA to implement packetization and block encoding (such as the block encoding taught by Shin) in the link layer of Yusairi, given this technique’s well-known use in well-known standards at the time, such as HDMI and Ethernet. Ex.1003, ¶218 (citing Ex.1010 at 40, Fig. 5-1); *id.* (citing Ex.1011, 98, 896, 928, Figs. 35-1, 35-1, 36.3). A POSITA also would have had a reasonable expectation of success in making this combination because it would have been a minor modification involving changes only to the encoding scheme used and would have been implemented on Yusairi’s existing link layer packetizing circuitry. Ex.1003, ¶218.

[68(b)] wherein the input words are indicative of application data and control bits; and

Yusairi in view of Shin renders obvious this limitation for the reasons discussed above at limitation [23(e)]. Ex.1003, ¶219.

[68(c)] encoding and transmission circuitry coupled to the packetizing circuitry and configured to encode the input words in accordance with a block code to generate a code word sequence of N-bit code words, where $N > M$,

Yusairi in view of Shin renders obvious this limitation.

As discussed at [68(a)], Yusairi discloses link-layer packetizing circuitry, and it would have been obvious to a POSITA to use Shin's block encoding methods with Yusairi's link-layer packetizing circuitry. Further, as also discussed for [68(a)], Yusairi generates packets and passes them to the trans module, which then transmits them to the physical layer with a request that they be transmitted out. Ex.1005, 54. Secs. 5, 5.1, 5.2. Thus, Yusairi discloses *transmission circuitry* that is *coupled to the packetizing circuitry* by also being located in the link layer. Ex.1003, ¶¶220-221.

In addition, Shin discloses encoding and transmission circuitry in a link layer that is coupled to packetizing circuitry and that performs Shin's block encoding discussed above for 68[a]. Specifically, Shin discloses an encoder 221 (*encoding circuitry*) and transmitter 231 (*transmission circuitry*) coupled to a packetizer 212, where the encoder of "[t]he link layer encodes the codes to be transmitted via its communications link into symbols ... and provides those symbols the [transmitter of] the physical layer" that "transmits the serialized symbols via the communications link." Ex.1004, [0076], [0077]; Ex.1003, ¶222. Shin further discloses encoding methods that *encode the input words in accordance with a block code to generate a*

code word sequence of N-bit code words, where $N > M$ for the reasons discussed above at limitation [2] of Grounds 1 and 2. Ex.1003, ¶222.

A POSITA would have been motivated to modify Yusairi's link layer by coupling Yusairi's link-layer packetizing circuitry with Shin's link-layer encoder and transmitter (*encoding and transmission circuitry*) that encode and transmit in block code that generates N-bit code words from M-bits, where $N > M$, for the reasons discussed above for limitations [2] and [6]. Ex.1003, ¶223. Further, a POSITA would have also been motivated to make this combination because, like Yusairi's link layer packetizing circuitry, Shin discloses link-layer encoding and transmission circuitry, which would have motivated a POSITA to make this combination. Ex.1003, ¶223. Further, a POSITA would have had a reasonable expectation of success given that Shin's encoder and transmitters are both in the link layer already, like Yusairi's packetizing circuitry. Ex.1003, ¶223.

Shin further discloses encoding methods that *encode the input words in accordance with a block code to generate a code word sequence of N-bit code words, where $N > M$ for the reasons discussed above at limitation [2] of Grounds 1 and 2. Ex.1003, ¶224.*

[68(d)] the code word sequence is indicative of at least one cell of the input words, the cell includes at least two of the control bits,

Yusairi in view of Shin renders obvious this limitation. As described with respect to limitations [1(f)], Yusairi discloses at least one packet (*cell*) that includes *at least two of the control bits*, because the transaction code bits and field type bits are within the same packet. Ex.1005, 54, Sec. 5.2. Shin discloses that the code word sequence is indicative of at least one cell of the input words, as described with respect to limitation [2] of Grounds 1 and 2. A POSITA would have been motivated to encode Yusairi's packets using Shin's block encoding techniques to render this claim element obvious for the same reasons discussed with respect to limitations [2] and [6]. Ex.1003, ¶225-227.

[68(e)] and said at least two of the control bits have multiple levels of communication protocol functionality,

Yusairi discloses this limitation for the reasons discussed with respect to [23(f)]. Ex.1003, ¶228.

[68(f)] wherein the encoding and transmission circuitry is configured to be coupled to a serial link and to transmit encoded data over the serial link,

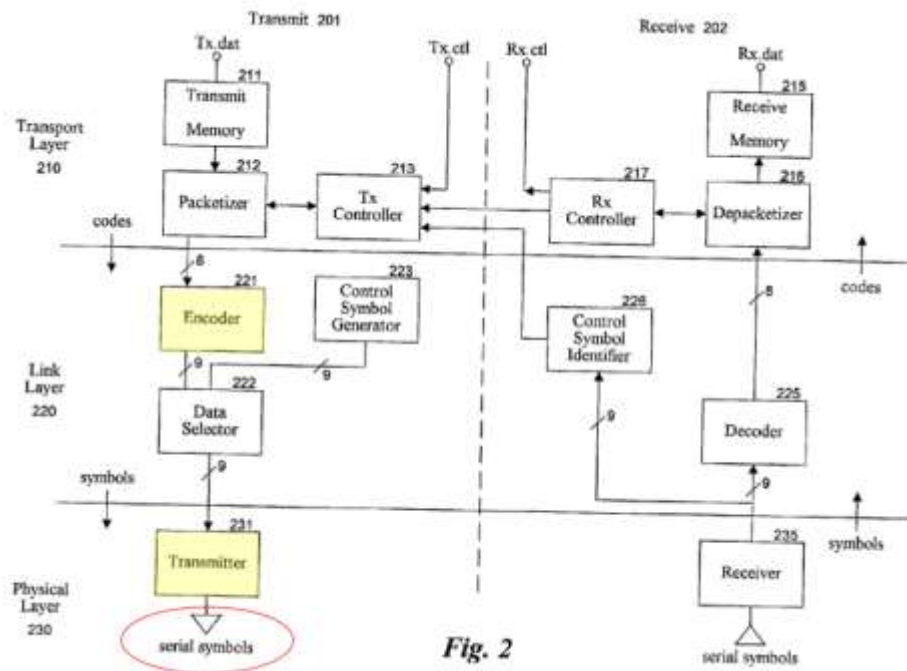
Yusairi discloses this limitation and Yusairi in view of Shin also renders obvious this limitation.

First, Yusairi discloses this limitation. As discussed for 68[c], Yusari discloses *encoding and transmission circuitry* through its disclosure that Yusairi's

stream_trans module performs encapsulation of packets and passes them to the trans module, which then transmits them to the physical layer with a request that they be transmitted out. Ex.1005, 54, Secs. 5, 5.1, 5.2; Ex.1003, ¶229-230.

Further, Yusairi discloses that its *encoding and transmission circuitry is configured to be coupled to a serial link and to transmit encoded data over the serial link*. Yusairi discloses that to send packet data a ‘Bus request’ must be performed by the link layer to the physical layer in which “[t]he Lreq terminal controlled by LLC sends serial service requests to PHY for taking the access of the serial-bus [serial link] for packet transmission” (*encoding and transmission circuitry is configured to be coupled to a serial link and to transmit encoded data over the serial link*). Ex1005, 52-52, Secs. 3.2 3.2.1. Ex.1003, ¶231.

In addition, Yusairi in view of Shin renders this limitation obvious. As discussed for [68(c)], Shin discloses *encoding and transmission circuitry* in the form of its encoder 221 and transmitter 231 that are *configured to be coupled to a serial link and to transmit encoded data over the serial link*. Specifically, as shown in Figure 2 of Shin, the encoder 221 is coupled to the transmitter 231, which is coupled to a “communications link” over which it transmits “serial symbols” (*coupled to a serial link and to transmit encoded data over the serial link*). Ex.1004, [0079], [0080], Fig. 2; Ex.1003, ¶232.



Ex.1004, Fig. 2 (annotated). A POSITA would have been motivated to combine Shin's encoder 221 and transmitter 231 (*encoding and transmission circuitry*) that are configured to be coupled to a serial link and to transmit encoded data over the serial link with Yusairi's link layer packetizing circuitry for the same reasons set forth above for [68(a)] and [68(c)]. Ex.1003, ¶233.

[68(g)] said encoded data including the code words of the code word sequence.

Yusairi in view of Shin renders obvious this limitation for the reasons discussed at limitation [2] of Ground 1 and 2. Ex.1003, ¶234.

A POSITA would have been motivated to use Shin's block encoding with Yusairi's link layer packetizing circuitry this limitation obvious for the reasons discussed with respect to limitations [2] and [6]. Ex.1003, ¶235.

12. Claim 70

[70] The transmitter of claim 68, wherein the block code is specified by a line code, the line code also specifies special characters, and the encoding and transmission circuitry is configured to generate the encoded data such that said encoded data are a sequence of at least one of the special characters and the code words of the code word sequence.

Yusairi in view of Shin renders obvious this limitation.

Shin discloses and renders obvious *the block code is specified by a line code, the line code also specifies special characters and that said encoded data are a sequence of at least one of the special characters and the code words of the code word sequence* for the reasons discussed with respect to [2] of Grounds 1 and 2. Ex.1003, ¶¶236-237.

Further, Yusairi discloses and Yusairi in view of Shin renders obvious *the encoding and transmission circuitry is configured to generate the encoded data* for the reasons discussed above with respect to [68(c)] and to transmit the encoded data as line code for the reasons discussed above with respect to [68(f)]. Ex.1003, ¶238.

A POSITA would have been motivated to use Shin's encoding method with Yusairi's link layer packetizing circuitry for the reasons discussed for 68[a] and 68[f], as well as for limitations [2] and [6]. Ex.1003, ¶239.

13. Claim 71

[71] The transmitter of claim 70, wherein none of the special characters of the encoded data matches a bit sequence of the code word sequence.

Yusairi in view of Shin renders obvious this limitation for the reasons discussed for limitation [25]. Ex.1003, ¶240.

It would have been obvious to a POSITA to modify Yusairi with Shin's teaching of special characters not matching bit sequences of the code word sequence so that the receiver can identify the special characters from among the transmitted code words and perform a function indicated by the special character, such as notifying a transmitter to stop communicating data using an XOFF special character. Ex.1004, [0079]; Ex.1003, ¶241.

14. Claim 72

[72] The transmitter of claim 70, wherein said at least one of the special characters is a link training character.

Yusairi in view of Shin renders obvious this limitation. The '307 Patent discloses that a special character called a "link training character" can be transmitted "[t]o establish symbol framing...." Ex.1001, 17:1-5, 17:22-23. Similarly, Shin teaches transmitting "synchronization primitives so that the receiving communications node can properly align with that synchronization primitive." Ex.1004, [0096]. A POSITA would have understood that the use of such a synchronization primitive establishes symbol framing to properly align the

transmitting node and receiving node, since the “node ... knows the primitive is correctly aligned on the symbol boundary.” *Id.*; Ex.1003, ¶¶242-243. Therefore, Shin’s synchronization primitive discloses *a link training character*. Ex.1003, ¶243. A POSITA would have understood a synchronization primitive is a *special character* because “[t]he synchronization primitives have a unique number of transitions that is not found in any control primitive or in-band symbol.” Ex.1004, [0101]; Ex.1003, ¶244.

A POSITA would have been motivated to combine Shin’s disclosure of a synchronization primitive acting as *a link training character* with Yusairi because doing so would allow for the SOPC Board B (acting as a receiver) to align symbol boundaries of transmitted encoded data, as taught by Shin. Ex.1004, [0096]; Ex.1003, ¶245.

15. Claim 74

[74] The transmitter of claim 70, wherein said at least one of the special characters is an idle character.

Yusairi in view of Shin renders this limitation obvious for the reasons discussed for limitation [2] of Ground 1 and 2. Ex.1003, ¶246.

A POSITA would have been motivated to use Shin’s IDLE primitives with Yusairi because doing so was a known way to provide the benefit of synchronizing communication nodes, such as the SOPC Boards, since Shin discloses “an IDLE

primitive may be a synchronization primitive,” that helps “a receiving communications node [to] properly align with [the] synchronization primitive” transmitted by a transmitting communications node.” Ex.1004, [0096], [0097]; Ex.1003, ¶247.

16. Claim 75

[75] The transmitter of claim 70, wherein said at least one of the special characters is a flow control character.

Yusairi in view of Shin renders obvious this limitation. Shin discloses this limitation. The '307 Patent states that an “idle character” can be for flow control. Ex.1001, 9:32-37. As such, a POSITA would have understood that Shin’s idle character is a flow control character because it can be used for flow control, as discussed for limitation [7] of Grounds 1 and 2. Ex.1003, ¶¶248-249.

As discussed above with respect to limitation [2], Shin discloses an IDLE primitive or symbol is a *special character*. Shin’s “IDLE primitive may be a synchronization primitive” that “maintain[s] proper synchronization” between transmitters and receivers. Ex.1004, [0097], [0105]. A POSITA, therefore, would have understood that Shin’s IDLE primitive is a special character for flow control when used for synchronization. Ex.1003, ¶249.

A POSITA would have been motivated to use Shin’s IDLE primitive acting as a *flow control character* with Yusairi because doing so would allow for the SOPC

Board A (acting as a transmitter) to synchronize with SOPC Board B (acting as the receiver) when transmitting encoded data, as taught by Shin. Ex.1004, [0096]; Ex.1003, ¶250.

E. CONCLUSION

Petitioner respectfully requests that the Challenged Claims be cancelled.

Respectfully submitted,

Dated: October 2, 2024

/Jason R. Mudd /
Jason R. Mudd
Lead Counsel
Reg. No. 57,700

X. MANDATORY NOTICES

A. Real Party-in-Interest

Pursuant to 37 C.F.R. § 42.8(b)(1), Petitioners hereby name Dell Inc., Dell Technologies Inc., HP Inc., and Lenovo (United States) Inc. as real parties-in-interest and, solely because it is named as a defendant in a co-pending district court case listed below, further identify Lenovo Group Ltd. as a real party-in-interest.

B. Related Matters

Pursuant to 37 C.F.R. § 42.8(b)(2), to the best knowledge of Petitioners, the '307 Patent is or was involved in the following cases:

Case Heading	Number	Venue	Filed
<i>Universal Connectivity Technologies Inc. v. HP Inc. f/k/a Hewlett-Packard</i>	1:23-cv-01177	W.D. Tex.	Sep. 28, 2023
<i>Universal Connectivity Technologies Inc. v. HP Inc.</i>	4:24-cv-04097	N.D. Cal.	Jul. 8, 2024
<i>Universal Connectivity Technologies Inc. v. Dell Techs. Inc. et al.</i>	1:23-cv-01506	W.D. Tex.	Dec. 12, 2023
<i>Universal Connectivity Technologies Inc. v. Lenovo Grp. Ltd.</i>	2:23-cv-00449	E.D. Tex.	Sep. 28, 2023

Petitioners are concurrently filing this Petition and IPR petitions for other asserted patents in IPR2024-01428, IPR2024-01478, IPR2024-01479, IPR2024-01480, IPR2024-01481, and IPR2024-01482, as shown below:

See:

U.S. Patent No.	Case No.
7,154,905	IPR2024-01428
8,680,712	IPR2024-01478
9,232,265	IPR2024-01479
9,852,103	IPR2024-01480
7,746,798	IPR2024-01481
7,856,520	IPR2024-01482

C. Lead and Back-up Counsel and Service Information

Lead Counsel:

Jason R. Mudd (Reg. No. 57,700)
Shook, Hardy & Bacon L.L.P.
2555 Grand Boulevard
Kansas City, MO 64108
Phone: 816-559-2437
Email: jmudd@shb.com

Back-up Counsel:

Lauren E. Douville (Reg. No. 77,145)
Shook, Hardy & Bacon L.L.P.
2555 Grand Boulevard
Kansas City, MO 64108
Phone: 816-559-0348
Email: ldouville@shb.com

Paul M. Vogel (Reg. No. 74,774)
Shook, Hardy & Bacon L.L.P.
2555 Grand Boulevard
Kansas City, MO 64108
Phone: 816-474-6550
Email: pvogel@shb.com

Brian W. Oaks (Reg. No. 44,981)
McDermott Will & Emery LLP

303 Colorado Street, Suite 2200
Austin, TX 78701
Phone: 512-726-2574
Email: boaks@mwe.com

Nick Schuneman (Reg. No. 62,088)
McDermott Will & Emery LLP
303 Colorado Street, Suite 2200
Austin, TX 78701
Phone: 512-726-2572
Email: nschuneman@mwe.com

Tessa Kroll (Reg. No. 79,327)
McDermott Will & Emery LLP
444 West Lake Street, Suite 4000
Chicago, IL 60606
TEL: 312-984-6973
Email: tkroll@mwe.com

Scott Weidenfeller (Reg. No. 54,531)
Covington & Burling LLP
One CityCenter,
850 Tenth Street, N.W.
Washington, DC 20001-4956
Phone: 202-662-5923
Fax: 202-778-5923
Email: sweidenfeller@cov.com

David A. Garr (Reg. No. 74,932)
Covington & Burling LLP
One CityCenter,
850 Tenth Street, N.W.
Washington, DC 20001-4956
Phone: 202-662-5923
Fax: 202-778-5923
Email: dgarr@cov.com

Please address all correspondence in this proceeding to lead and back-up counsel at the following e-mail addresses:

HP-UCT-IPR@shb.com
DL-DELL-UCT-MWE@mwe.com
Lenovo-UCT-IPR@cov.com

Petitioners consent to service in this proceeding by e-mail at the addresses above.

CERTIFICATE OF WORD COUNT

Pursuant to 37 C.F.R. § 42.24(d), Petitioner hereby certifies, in accordance with and reliance on the word count provided by the word-processing system used to prepare this Petition, that the number of words in this paper is 13,989. Pursuant to 37 C.F.R. § 42.24(d), this word count excludes the table of contents, table of authorities, mandatory notices under § 42.8, certificate of service, certificate of word count, appendix of exhibits, and any claim listing.

Dated: October 2, 2024

/Jason R. Mudd /
Jason R. Mudd
Lead Counsel
Reg. No. 57,700

CERTIFICATE OF SERVICE

The undersigned certifies that, in accordance with 37 C.F.R. § 42.6(e) and 37 C.F.R. § 42.105, service was made on Patent Owner at the Correspondence Address of record for the '307 Patent listed in Patent Center as detailed below:

<i>Date of service</i>	October 2, 2024
<i>Manner of service</i>	UPS OVERNIGHT
<i>Documents served</i>	Petition for <i>Inter Partes</i> Review Under 35 U.S.C. § 312 and 37 C.F.R. § 42.104 of U.S. 7,187,307; Petitioner's Exhibit List; and Exhibits Ex.1001-Ex.1024.
<i>Persons served</i>	60974 - GIRARD & EQUITZ LLP 1539 TARAVAL ST. SUITE 202 SAN FRANCISCO, CA 94116 UNITED STATES

With a courtesy copy of the Petition and Exhibits sent on October 2, 2024 via e-mail to Patent Owner's litigation counsel of record:

Brett E. Cooper
bcooper@bc-lawgroup.com
Seth Hasenour
shasenour@bc-lawgroup.com
Jonathan Yim
jyim@bc-lawgroup.com
Drew B. Hollander
dhollander@bc-lawgroup.com

/ Jason R. Mudd /
Jason R. Mudd
Lead Counsel for Petitioner
Reg. No. 57,700

CLAIM LISTING APPENDIX

U.S. Pat. No. 7,187,307

Designation	Claim Language
Claim 1	
[1(pre)]	A communication system, including:
[1(a)]	a first endpoint;
[1(b)]	a second endpoint, including physical and link layer circuitry and higher level circuitry coupled to the physical and link layer circuitry; and
[1(c)]	a serial link between the first endpoint and the second endpoint,
[1(d)]	wherein the first endpoint is configured to generate encoded data in accordance with a line code, and to transmit the encoded data over the link to the second endpoint, and
[1(e)]	wherein the physical and link layer circuitry of the second endpoint is coupled to the link and configured to decode the encoded data
[1(f)]	to recover application data and control bits,
[1(g)]	to perform at least one link level function in response to at least one of the control bits, and
[1(h)]	to assert at least one control signal indicative of at least another one of the control bits to the higher level circuitry, and
[1(i)]	wherein the higher level circuitry is configured to perform at least one higher level function in response to the at least one control signal.

Designation	Claim Language
Claim 2	
2	The system of claim 1, wherein the line code specifies a block code and special characters, the first endpoint is configured to encode M-bit input words in accordance with the block code to generate a code word sequence of N-bit code words, where $N > M$, the code word sequence is indicative of at least one cell of the input words, and the encoded data are a sequence of at least one of the special characters and the code words of the code word sequence.

Designation	Claim Language
Claim 6	
6	The system of claim 2, wherein said at least one of the special characters is an idle character.

Designation	Claim Language
Claim 7	
7	The system of claim 6, wherein the physical and link layer circuitry is configured to perform simplex link flow control by inserting idle characters between subsequences of the code word sequence, where each of the subsequences is indicative of a different cell of the input words.

Designation	Claim Language
Claim 23	
[23(pre)]	A communication system, including:
[23(a)]	a first endpoint;
[23(b)]	a second endpoint; and
[23(c)]	a serial link between the first endpoint and the second endpoint,
[23(d)]	wherein the first endpoint is configured to encode M-bit input words in accordance with a block code to generate a code word sequence of N-bit code words, where $N > M$,
[23(e)]	the code word sequence is indicative of at least one cell of the input words, the input words of each said cell are indicative of application data and control bits,
[23(f)]	the control bits have multiple levels of communication protocol functionality, and
[23(g)]	the first endpoint is configured to transmit encoded data including the code words of the code word sequence over the link to the second endpoint.

Designation	Claim Language
Claim 25	
25	The system of claim 23, wherein the block code is specified by a line code that also specifies special characters, the encoded data are a sequence of at least one of the special characters and the code words of the code word sequence, and none of the special characters of the encoded data matches a bit sequence of the code word sequence.

Designation	Claim Language
Claim 31	
31	The system of claim 23, wherein each of the first endpoint and the second endpoint is a transceiver, and the second endpoint is configured to generate additional encoded data in response to input words in accordance with the line code and to transmit the encoded data over the link to the first endpoint.

Designation	Claim Language
Claim 53	
53[(pre)]	A receiver for receiving encoded data that have propagated over a serial link, the encoded data having been generated in accordance with a line code, said receiver including:
53[(a)]	physical and link layer circuitry, coupled and configured to receive and decode the encoded data
53[(b)]	to recover application data and control bits
53[(c)]	to perform at least one link level function in response to at least one of the control bits; and
53[(d)]	higher level circuitry coupled to the physical and link layer circuitry, wherein the physical and link layer circuitry is coupled and configured to assert at least one control signal indicative of at least another one of the control bits to the higher level circuitry,

Designation	Claim Language
Claim 53	
53[(e)]	and the higher level circuitry is configured to perform at least one higher level function in response to the at least one control signal.

Designation	Claim Language
Claim 68	
68[(pre)]	A transmitter including physical and link layer circuitry, wherein the physical and link layer circuitry include:
68[(a)]	packetizing circuitry configured to generate M-bit input words,
68[(b)]	wherein the input words are indicative of application data and control bits; and
68[(c)]	encoding and transmission circuitry coupled to the packetizing circuitry and configured to encode the input words in accordance with a block code to generate a code word sequence of N-bit code words, where $N > M$,
68[(d)]	the code word sequence is indicative of at least one cell of the input words, the cell includes at least two of the control bits,
68[(e)]	and said at least two of the control bits have multiple levels of communication protocol functionality,
68[(f)]	wherein the encoding and transmission circuitry is configured to be coupled to a serial link and to transmit encoded data over the serial link,
68[(g)]	said encoded data including the code words of the code word sequence.

Designation	Claim Language
Claim 70	
70	The transmitter of claim 68, wherein the block code is specified by a line code, the line code also specifies special characters, and the encoding and transmission circuitry is configured to generate the encoded data such that said encoded data are a sequence of at least one of the special characters and the code words of the code word sequence.

Designation	Claim Language
Claim 71	
71	The transmitter of claim 70, wherein none of the special characters of the encoded data matches a bit sequence of the code word sequence.

Designation	Claim Language
Claim 72	
72	The transmitter of claim 70, wherein said at least one of the special characters is a link training character.

Designation	Claim Language
Claim 74	
74	The transmitter of claim 70, wherein said at least one of the special characters is an idle character.

Designation	Claim Language
Claim 75	
75	The transmitter of claim 70, wherein said at least one of the special characters is a flow control character.