United States District Court,
D. Delaware.

BELL COMMUNICATIONS RESEARCH, INC,
Plaintiff.
v.
FORE SYSTEMS, INC,
Defendant.

No. CIV.A. 98-586-JJF

Aug. 29, 2000.

Owner of patents for multiplexing and demultiplexing digital data streams sued competitor for infringement. Construing claims language, the District Court, Farnan, J., held that: (1) preamble for multiplexing patent did not require that method steps all be performed at single transmitting device; (2) bit streams involved could be created in either serial or parallel configuration; (3) multiplexing patent required two or more empty data frames to be filled at same time by different data sources; (4) preamble for demultiplexing patent limited it to taking of multiplexed data frames, interleaved without gaps or pauses, and separating them into their original, single, data frames, each of which had exactly same format; and (5) demultiplexing patent required that bit patterns from two or more adjacent reconstructed bytes be compared with bit patterns known to have been present in two or more adjacent bytes in each of the contributory frames which had been multiplexed to form received bit stream.

Claims construed.

4,835,768, 4,893,306. Construed.

Richard K. Herrmann, Esquire, Mary B. Matterer, Esquire, Dale R. Dube, Esquire of Blank Rome Comisky Mccauley L.L.P., Wilmington, Delaware. Of counsel: Donald Dunner, Esquire, Vincent P. Kovalick, Esquire, Richard H. Smith, Esquire, Frank A. DeCosta, III, Esquire, Houtan K. Esfahani, Esquire of Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P., Washington, D.C. Attorneys for Plaintiff and Counter-defendant, Bell Communications Research, Inc.

William J. Marsden, Jr., Esquire of Fish & Richardson P.C., Wilmington, Delaware. Of counsel: John E. Gartman, Esquire, Christopher S. Marchese, Esquire, Amar L. Thakur, Esquire, Todd G. Miller, Esquire of Fish & Richardson P.C., San Diego, California. Attorneys for Defendant and Counter-plaintiff, Fore Systems, Inc.

OPINION

FARNAN**, District Judge.**

This action was brought by Plaintiff, Bell Communication Research, Inc. ("Bellcore") against Defendant Fore, Systems, Inc. ("FORE") alleging infringement of United States Patent Nos. 4,893,306 (the " '306 Patent"); 4,835,768 (the " '768 Patent"), 4,740,954 (the " '954 Patent") and 4,706,080 (the " '080 Patent"). The issue currently before the Court is the claim construction of the patents in suit. The parties briefed their respective positions on claim construction, and the Court held a *Markman* hearing on March 23, 2000. Thereafter, on August 1, 2000, the parties stipulated to the dismissal of Bellcore's claims under the '080 and '954 Patents (D.I.340) leaving only the disputed portions of the '306 and '768 Patents for the Court's claim construction. This Opinion presents the Court's constructions of the disputed terms in the '306 and '768 Patents. FN1

FN1. For ease of reference, the Court has attached the full text and accompanying figures of the '306 and '768 Patents to this Opinion.

BACKGROUND

I. Introduction to the Technology Generally

The '306 and '768 Patents relate to telecommunications technology. A telecommunicationsnetwork can accept input from various sources like speech from a telephone, data from a computer, or a video signal from a camera. In today's telephone network system, voice signals are converted into "digital" signals by analog-digital converters. A digital signal represents information in a binary form or "bit." A bit can have a value of "1" or "0." Bits are typically transmitted between telephone switches in parallel groupings of eight bits known as "bytes."

Like telephones, computers are also connected by networks which utilize this transmission process. Small localized networks like those used by single business entities are called "Local Area Networks" or "LANs." LANs can be connected together by computers called "routers" or "bridges" to form larger networks typically referred to as "WANs" or "Wide Area Networks."

II. The Patents

The '306 and '768 Patents relate to transmission systems. Specifically, they are concerned with how a series of bits traveling down a physical wire are grouped and packaged by the sending source for transmission and how they are ungrouped and unpackaged by the receiving source.

A. *The* '306 Patent

The '306 patent describes a digital network transport system known as Dynamic Time Division Multiplexing ("DTDM"). In a DTDM network, the fundamental unit of data transport is known as a frame. Each frame contains two fixed length fields, an overhead and a payload. The overhead field contains information such as the empty/full status of the frame and information related to timing. The payload field of the frame may be filled with a data packet, which contains information and a header field. The header field serves a similar purpose to the address on a mailing envelope. Stated another way, each occupied frame contains a transmission overhead field, a header field, and an information field.

Figure 2 of the '306 Patent represents the assembly of the DTDM bit stream. A "train" of DTDM frames 10

with empty payload fields is generated. This train has a transmission or bit rate that serves as the basic backbone transmission rate for the system. The assembler 3 inserts data from different sources known as tributaries into the train 10. In Figure 2, items 5, 7 and 9 represent three different tributaries or information input streams, each being transmitted from a different source and at a different rate of speed. For example, stream 5 might originate from a telephone, stream 7 from a computer, and stream 9 from a video camera. Before this information can be inserted into stream 10, the packetizers (11, 13, and 15) each take their respective input stream and break them up into fixed length packets of data and attach a packet header (H). The completed process is shown in Figure 2 by items 17, 19 and 21.

The packets that comprise streams 17, 19 and 21 are inserted into the empty payload fields of the frames in stream 10. The end result, represented by stream 12, is a stream of data at a single transmission rate that has multiplexed information received from three different sources at three different transmission rates.

B. *The* '768 Patent

The '768 Patent relates to a circuit and technique for recognizing and identifying information transported in a transmission bit stream. The '768 Patent discusses the invention in terms of fiber optic transmission systems. Fiber optic transmission systems utilize optical fibers to carry great amounts of information at the speed of light. As explained previously, this information is transmitted in the form of bits. In order for computers to understand how to read the bits it receives as a bit stream, communications designers organize the bits into predetermined patterns or structures. The patterns or structures for organizingbits and the rules for interpreting them are known as "protocols."

One protocol used for fiber optics transmission is called SONET (Synchronous Optical Network). The basic SONET structure or "frame" consists of nine rows of ninety data bytes. Of the ninety bytes in each row, 3 bytes are transport overhead information and 87 bytes are payload information. The first two bytes of transport overhead, identified in the '768 Patent as F1 and F2, are framing bytes which can be recognized by a receiver to synchronize the receiving circuits to the SONET frame structure. The F1 and F2 bytes have distinct bit patterns which are always the same.

A SONET frame is transmitted row by row as a sequential bit stream beginning with the first F1 byte in row one through the last payload byte in row one, followed by the first overhead byte in row two through the last payload byte in row two, and so forth. The process continues byte by byte and row by row until the entire frame has been transmitted.

Although the basic SONET frame ("STS-1") permits the fast transmission of information, this transmission is not fast enough for some applications. To facilitate even faster transmission of information, the SONET designers wrote protocols for forming larger frames which are basically multiples of the basis SONET STS-1 frame. For example, an STS-3 frame is formed by combining three STS-1 frames, and an STS-12 frame is formed by combining four STS-3 frames. Although an STS-3 frame carries three times as much information at three times the bit rate of an STS-1 frame, the STS-3 frame is transmitted at the same frame rate. The creation of these larger frames using multiples of the basic STS-1 frame is known as the SONET multiplexing hierarchy.

The '768 Patent relates to a technique for recognizing and reconstructing the bytes from a bit stream generated in accordance with the SONET hierarchy. When a computer receives a bit stream, it must reconstruct and identify each frame in order to retrieve the data being transported. For example, the

computer must identify the F1 byte to synchronize itself to the beginning of the frame. Once the beginning of the frame is located, the frame boundaries can be located by counting bytes.

The '768 Patent provides a fast method for recovering bytes and identifying and synchronizing to frame boundaries. The invention in the '768 Patent provides for a shift register that accumulates the serially-received bits and holds a byte of data. A bit comparator compares the accumulated bits with the known bit pattern for the F1 byte. When the accumulated bits match the known pattern for the F1 byte the comparator generates a signal. Circuits respond to the first signal by outputting the F1 byte and each subsequently received byte. The sequence of byte signals generated by the comparator represents the reconstructed bytes of the originally transmitted frame.

In order to ascertain the frame boundaries, additional comparing circuits compare output bytes to the known byte patterns for the F1 and F2 bytes. The circuitry is basically looking for a transition from F1 type bytes to F2 type bytes or the sequence F1, F2, F2. When the circuitry detects the change a second signal ("FP") is generated. This second signal is the benchmark from which the frame boundaries are determined. Because the receiver knows the STS-N level of the incoming bit stream, it also knows at what byte of the frame the transition from F1 to F2 occurred. The receiver then knows when the first byte of the next frame will be received by counting the bytes following the F1 to F2 transition.

When complete frame synchronization is achieved, the comparator is disabled by a toggle, so that the system won't shift the byte boundaries by any subsequent errors in the F1 pattern. If the second signal does not occur for a number of consecutive frames, an out of frame signal ("OOF") is generated and the toggle turns the comparator back on to search for the F1 pattern again.

DISCUSSION

I. The Legal Principles of Claim Construction

[1] [2] [3] [4] [5] Claim construction is a question of law. Markman v. Westview Instruments, Inc., 52 F.3d 967, 977-78 (Fed.Cir.1995), *aff'd,* 517 U.S. 370, 388-90, 116 S.Ct. 1384, 134 L.Ed.2d 577 (1996). When construing the claims of a patent, a court considers the literal language of the claim, the patent specification and the prosecution history. Markman, 52 F.3d at 979. A court may consider extrinsic evidence, including expert and inventor testimony, dictionaries, and learned treatises, in order to assist it in construing the true meaning of the language used in the patent. Id. at 979-80 (citations omitted). A court should interpret the language in a claim by applying the ordinary and accustomed meaning of the words in the claim. Envirotech Corp. v. Al George, Inc., 730 F.2d 753, 759 (Fed.Cir.1984). However, if the patent inventor clearly supplies a different meaning, the claim should be interpreted accordingly. Markman, 52 F.3d at 980 (noting that patentee is free to be his own lexicographer, but emphasizing that any special definitions given to words must be clearly set forth in patent). If possible, claims should be construed to uphold validity. In re Yamamoto, 740 F.2d 1569, 1571 & n. * (Fed.Cir.1984) (citations omitted).

[6] Before turning to the patents at issue in this dispute, the Court is compelled to address what it perceives to be a potential problem regarding the claim construction issues in this case. Throughout their briefs, the parties raise terms or phrases for claim construction which are either not responded to by the opposing party or are rebutted by the opposing party without sufficient explanation and/or without advancing a counter-proposal for construction. In its Opposition Markman Brief, FORE acknowledges this issue and indicates that it will not engage in claim interpretation for the sake of "achieving linguistic purity and absolute hypothetical precision on every word." (D.I. 153 at 1). FORE also states that "[t]o the extent that Bellcore's

definitions do not affect the issues in this case, they are an advisory and needless exercise in claim interpretation-and should be rejected." (D.I. 153 at 1-2). The Court agrees that claim interpretation should involve only genuinely disputed terms that impact infringement or validity issues. This having been said, the Court declines to adopt constructions for terms and/or phrases raised by only one party and either not addressed by the other party or not responded to with sufficient information and/or a counter-proposal for claim construction. However, if after joint consultation between the parties, one party still genuinely believes that additional terms or phrases impact infringement or validity and require construction, the Court will require the party to submit a letter memorandum, no more than 3 pages in length (with customary margins and font size), stating the terms in need of construction, the proposed constructions, and the reasons for the construction. The opposing party is then required to submit a response letter, no more than 3 pages in length (with customary margins and font size), indicating their position, i.e. whether they concede to the definitions and if not, offering alternative proposed constructions and the reasons for the proposed constructions. In permitting the parties this opportunity, the Court wishes to make clear that these letter memoranda shall not reargue constructions which have been decided in this Opinion and shall not raise terms or phrases which were not previously raised by a party in the previous sets of claim construction briefing. With this understanding, the Court will proceed to construe the disputed terms of the '306 and '768 Patents.

II. The Meaning Of The Disputed Terms of the '306 and '768 Patents

A. *The* '306 Patent

Bellcore asserts Claims 1, 3 and 4 of the '306 patent. The Court will address the disputes relevant to each claim in turn.

1. Claim 1 of the '306 Patent

In full, Claim 1 of the '306 Patent provides:

A method for simultaneously transmitting data from sources having different bit rates in a telecommunication network comprising the steps of:

generating a bit stream comprising a sequence of frames, each of said frames including a transmission overhead field containing frame timing information and an empty payload field, and

filling the empty payload fields in said frames with data in packetized format from a plurality of sources which have access to the bit stream including circuit or packet sources, such that data in packetized format from any of said sources is written into any available empty payload field of any of said frames for transmitting data from each of said sources at its own desired bit rate via said bit stream and for transmitting data from said plurality of sources simultaneously via said bit stream.

The Court will examine each of the disputed terms and phrases below.

a. A method for simultaneously transmitting data from sources having different bit rates in a telecommunication network comprising the steps of:

[7] Bellcore contends that the language "[a] method for simultaneously transmitting data from sources having different bit rates in a telecommunication network" means that the method steps are performed at one

transmitting device. (D.I. 142 at 24-25; D.I. 144 at 10). According to Bellcore, the transmitting device simultaneously transmits data from more than one source in the telecommunications network connected to that transmitting device. In support of its position, Bellcore directs the Court to the conceptual illustrations depicted in Figures 2 and 3 of the '306 Patent. For example, Bellcore points out that the DTDM assembler 32 shown in Figure 3 performs both the generating and filling steps of the method and transmits frames on transmission line 62, which is part of a telecommunications network. In addition, Bellcore relies on language in the specification which provides, for example, that "the DTDM system, packet and circuit traffic can be multiplexed through the same multiplexer." ('306 Patent, col. 4, lines 65-66; col. 5, lines 13-15).

In addition to the figures and language of the specification, Bellcore also directs the Court to the prosecution history of the '306 Patent. According to Bellcore, the applicants amended Claim 1 of the '306 Patent to distinguish prior art known as the Shikama et al. reference. (D.I. 143, Ex. J at A114-A117). Bellcore contends that in the Shikama reference frames are generated in one device in one network and data added to those frames at other devices in the network, but in the '306 Patent the method steps must be performed at one transmitting device which simultaneously transmits data from more than one source in the telecommunications network connected to that transmitting device.

In response to Bellcore's proffered interpretation of the preamble language as requiring the methods steps to be performed at one transmitting device, FORE contends that neither the claim language nor the specification supports Bellcore's attempt to import the limitation of "one transmitting device" into the claim. In other words, FORE does not limit the language to require the acts to be performed at one transmitting device. (D.I. 153 at 2-4). FORE points out that the only conceivable structure mentioned in the plain language of this clause is the word "sources" and that this word says nothing about only "one transmitting device." FORE also contends that the figures shown in the patent specification show multiple devices and not one device. Thus FORE's proffered construction of this phrase is that the acts simply be performed in accordance with all the recited steps in the patent.

After reviewing the claim language, specification and prosecution history of the '306 Patent in light of the parties' respective positions, the Court agrees with FORE's interpretation of this language. Although the specification suggests that the method steps listed after the preamble *may* be performed using a single transmitting device, there is nothing in the claim language, specification or prosecution history *requiring* that the method be accomplished by a single transmitting device. For example, the language of the claim does not refer to a single device. Moreover, the very language cited by Bellcore in the specification provides that the DTDM system, packet and circuit traffic "can" or "may" be multiplexed through the same multiplexer, not that it "must" be multiplexed through the same multiplexer. Accordingly the Court construes the preamble language "[a] method for simultaneously transmitting data from sources having different bit rates in a telecommunication network comprising the steps of" to require that the acts be performed in accordance with the subsequently recited steps in the claim, and the Court declines to limit this phrase to require the method steps to be performed at one transmitting device.

b. generating a bit stream

[8] The crux of the parties' dispute concerning the phrase "generating a bit stream," is whether this language encompasses a stream in only serial configuration or a stream in either serial or parallel configuration. Bellcore contends that this language means generating bits or bytes at particular time intervals. (D.I. 142 at 25). Thus, according to Bellcore, the "bit stream" can be either a serial bit stream or a parallel bit stream that is "byte wide." In support of its position, Bellcore directs the Court to the preferred embodiment shown

in Figure 12 of the '306 Patent. (D.I. 151 at 7-8; D.I. 142 at 25-27).

Figure 12 illustrates the framer circuitry performing the generating step by originating and processing data as a parallel bit stream (in byte format-multiple bits at one time). Bytes are transmitted between framers in a serial bit stream. In the language of the specification

The framer unit 200 may also be utilized to generate a chain of empty DTDM packets [sic] (see e.g., framer 52 in FIG. 4). In this case the serial input 202 and associated serial-to-parallel converter 212 are not utilized. Instead, the control 210 applies a periodic signal to tristate 222 so that a frame alignment word is periodically read from *frame byte ROM 224* and *transmitted via bus 219 to parallel-to-serial converter 216* and serial output 206 so as to define a train of empty DTDM frames.

('306 Patent col. 16, line 63-col. 17, line 2) (emphasis added).

An examination of Figure 12 shows a single line used where serial data is present, for example item 206, and a double line used where parallel data is represented, for example item 219. Therefore, Bellcore argues, to construe the phrase generating a bit stream to encompass only a serial configuration would exclude the preferred embodiment set out in Figure 12. (D.I. 151 at 7).

In response to Bellcore's interpretation, Fore contends that "generating a bit stream" means creating a serial configuration of bits, or stated another way, a continuous line of bits sent one at a time from one point to another. To this effect, FORE contends that a "bit stream" is not a "byte stream." (D.I. 153 at 5-6; D.I. 146 at 10-11). In support of its position, FORE directs the Court to Figure 3 of the '306 Patent and its accompanying text in the specification which provides that:

the DTDM bit stream leaves the *serial* data output of framer (sdo) of framer unit 52 and enters the *serial* data input of the top most framer unit 53. The DTDM bit stream leaves the topmost framer 53 via its *serial* data output (sdo). The DTDM bit stream then enters the *serial* data input (sdi) of each succeeding framer unit and leaves via the *serial* data output (sdo) of each framer unit.

('306 Patent, col. 8, lines 62-66 (emphasis added)).

In addition to Figure 3, Fore also contends that the preferred embodiment in Figure 12 and its accompanying text supports its claim interpretation. Fore contends that while it is possible that bytes may be used internally in transmission equipment (D.I. 146 at 10), the frames generated and sent over transmission lines are in a serial bit configuration, as evidenced by the fact that the portion of the specification describing Figure 12 states that the empty DTDM frames are defined by sending data to parallel-to-serial converter 216 and then to serial output 206.

Additionally, FORE directs the Court to that portion of the specification describing bit stream rates. According to FORE, the specification describes bit stream rates in terms of megabits per second, confirming that a bit stream is a serial configuration of bits and not a parallel byte wide configuration. (D.I. 153 at 5, citing '306 patent, col. 2, lines 25-35; col. 5, lines 13-22).

The Court disagrees with FORE's interpretation that the phrase "generating a bit stream" is limited to a serial configuration of bits. First, there is nothing in the plain language of the claim limiting a "bit stream" to a "serial bit stream." Indeed, the claim does not utilize the modifying or limiting word "serial" in

describing the "bit stream." While it is true that the language used is "bit stream" and not "byte stream," it is also true that a byte is composed of bits. Therefore, the Court cannot conclude that the use of the word bit excludes eight bits in parallel, otherwise known as a byte.

With regard to FORE's argument that the specification refers to bit stream rates in terms of megabites per second, the Court is unpersuaded. The lines cited by FORE refer to rates measured in both Mb/sec and Mbit/sec, suggesting that Mb/sec might well mean megabytes per second.

As for Figure 12 and its accompanying text, the Court concludes that these references support Bellcore's position that the bit stream generated internally in the framer circuit begins in a parallel state and is then converted to a serial state before transmission. Accordingly, the Court concludes that the phrase "generating a bit stream" encompasses the creation of either serial or parallel bit streams.

c. frame timing information

[9] Bellcore argues that "frame timing information" should be construed to mean frame alignment information. (D.I. 144 at 11; D.I. 142 at 27; D.I. 151 at 8). According to Bellcore, the frame timing information permits a receiver to identify the start of a frame and synchronize to the frame boundaries. (D.I. 142 at 27). In support of its proposed construction, Bellcore cites language from the specification stating: "[t]he overhead field includes, for example, *a frame alignment word for frame timing* and the empty/full status of the frame" ('306 Patent, col. 4, lines 52-54 (emphasis added)) and "[t]he following information may be available in the overhead field of every frame, *frame alignment word for frame timing* ...." ('306 Patent, col. 6, lines 61-64 (emphasis added)).

Fore contends that "frame timing information should be construed to mean one or more bits that indicate the beginning of a frame." (D.I. 144 at 12; D.I. 146 at 11). Fore characterizes the question before the Court as whether "frame timing information" must be more than one bit, as Bellcore contends, or may it be one or more bits, as FORE contends. (D.I. 146 at 11). As for Bellcore's reliance on the specification's use of the phrase "frame alignment word" in discussing "frame timing," FORE contends that a "word" is composed of one or more bits just as a "word" in the English language is composed of one or more letters. (D.I. 146 at 11).

Examining the claim language and specification in light of the parties' arguments, the Court believes it is evident that there is a correlation between frame timing and frame alignment word. Indeed, based on FORE's argument, it appears that FORE agrees that the frame timing information has to do with alignment. Thus, as FORE contends, the question is whether the frame timing information must be more than one bit. The Court is not persuaded by FORE's argument that a "word" contemplates one or more bits just as "word" in the English language contemplates one or more letters. (D.I. 146 at 11). In the IEEE Standard Dictionary of Electrical and Electronic Terms, 6th Edition, there are sixteen different definitions of "word," all of which require more than one bit. (D.I. 152, Exh. M at A143-44). Accordingly, the Court concludes that "frame timing information" means frame alignment information comprised of more than one bit.

d. filling the empty payload fields in said frames with data in packetized format

[10] According to Bellcore, an "empty payload field" means "bits or time periods representing an absence of source data to be transmitted." (D.I. 144 at 11). Consistent with its definition of "empty payload field," Bellcore contends that "filling the empty payload fields" refers to outputting source data when it is available during a payload field interval. In support of its position, Bellcore directs the Court to Figure 12 of the '306

Patent and that portion of the specification which provides: " *if the particular DTDM frame is empty and data is available* at the parallel input, a signal is applied by the control to the tristate device to enable the data to be inserted into the particular DTDM frame via bus before it leaves the framer unit." ('306 Patent, col. 16, lines 49-55 (emphasis added)).

In response to Bellcore's proposed construction, FORE contends that an "empty payload field" means that a frame's payload has zero data in it. (D.I. 144 at 12). Consistent with its proposed construction of the term "empty payload field," FORE contends that "filling the empty payload fields in said frames" requires two steps: (1) that a complete empty frame is first created; and (2) after creation of the empty frame, the frame's payload is 100% filled with a packet. FORE contends that its construction is supported by the plain and ordinary meaning of the words used in the claim, as well as by the specification and prosecution history of the '306 Patent. For example, FORE points out that the language of Claim 1 contemplates generating multiple empty frames with frame timing information that can subsequently be filled with packets. FN2 FORE also points out that Figure 2 of the '306 Patent shows a generated train of empty frames entering the DTDM assembler. In addition, the language accompanying Figure 2 provides: "[A] train 10 of DTDM frames with empty payload fields is generated" and "[e]ach of the frames in the train 10 has an occupied transmission overhead field (T)." ('306 Patent, col. 7, lines 27-28). FORE further points out that Figure 4 and its accompanying text illustrates how the train of empty frames is generated in the first instance.The text provides: "The topmost framer 52 in Figure 4 does not have any input service connected to it. It generates the train of empty DTDM frames which are sent to the following framers 53." ('306 Patent, col. 9, line 59-61).

FN2. Fore relies on that portion of Claim 1 providing:
generating a bit stream comprising a *sequence* of frame *s,* each of said frame *s* including ... frame timing information and an empty payload field, and filling the empty payload field *s* in said frame *s* with data in packetized format....

('306 Patent, col. 17, lines 47-52 (emphasis added)).
In addition to the claim language and specification, FORE also directs the Court to the prosecution history of the '306 Patent. FORE contends that Bellcore distinguished prior art known as the Baran patent by stressing that the '306 Patent requires the generation of a sequence of empty frames first, and then filling the frames. Distinguishing Baran during the prosecution of the '306 Patent, Bellcore explained:

[T]he transmission bit streams are formed entirely differently in the claimed invention in Baran et al. Thus, as indicated above, the transmission stream of the claimed invention is formed by *first* generating a bit stream comprised of *frames with empty payload fields*. Data from a plurality of sources which have access to the transmission stream are packetized. The packets are *then* inserted into *the empty payload fields of the frames*. The Baran reference in no way discloses the formation of a transmission bit stream by generating a *sequence* of frame *s* with empty payload field *s* and picking up packets from a plurality of sources to fill the payload field *s*.

(D.I. 147, Ex. 7 at FSI001338 (emphasis added)). Similarly, Bellcore explained the '306 Patent during the prosecution with the following analogy:

The *stream of empty frames* may be analogized to a *train of empty freight cars*. The *empty freight cars are filled with data in packetized format* from various sources *which have access to the train of freight cars*. The

train, with its *now filled freight cars,* transmits the data to remote locations.

(D.I. 147, Ex. 7 at FSI001333 (emphasis added)).

After reviewing the parties' arguments in the context of the claim language, specification and prosecution history of the '306 Patent, the Court agrees with FORE's construction of the phrase "filling the empty payload fields in said frames with data in packetized format." The grammatical structure of the claim language confirms that multiple empty payload fields are generated, and then the empty payload fields are detected and then filled with packets. Stated another way, the frames cannot be filled until after they are generated and empty frames are detected. Indeed, the Court believes that this interpretation is consistent with Figures 2 and 4 of the patent specification and consistent with the position stressed by Bellcore during the prosecution of the '306 Patent. Accordingly, the Court construes the term "empty payload field" to mean that a frame's payload has zero data in it. The Court further construes "filling the empty payload fields" to mean that first the empty frames must be generated, and second the frames' empty payloads are filled with data.

e. data in packetized format from any of said sources is written into any available empty payload field of any of said frames

[11] In disputing the meaning of this phrase, Bellcore contends that this phrase means that packetized data can be placed into the payload field of any frame interval whenever a complete data packet becomes available. To this effect, Bellcore contends that the circuit does not wait for a particular frame or predetermined time to output the packet. Bellcore relies on Figures 4 and 12 of the '306 Patent in support of its argument. (D.I. 142 at 29).

In response to Bellcore's interpretation, FORE contends that this language means that packets are only put in frames that have zero data in their payload. FORE directs the Court to the "Summary of the Invention" section of the specification, which provides:

Illustratively, a DTDM multiplexer may be used to merge traffic from three different communications sources or tributaries into a single DTDM bit stream.... The available frames are shared by the three tributaries by giving higher priority to the circuit tributary, and allowing the voice and graphics tributaries to contend on a first-come, first-served basis. The circuit tributary *seizes one out of every three empty frames passing by* .... In this case the voice tributary will on average seize one out of every 2,160 frames. Similarly, at a rate of 1 Megabit per second, the graphics tributary will fill one frame out of 150. In this way, three diverse data streams are multiplexed into a single bit stream.

('306 Patent, col. 5, lines 13-38 (emphasis added)). According to FORE, this description of the invention makes it clear that each source detects and then "seizes" empty frames. FORE contends that once a packet is inserted into a frame, it is no longer empty so other sources will not seize the filled frame. Rather, if another source has a packet to send, it will insert its packet into the next "available empty frame."

The Court agrees with FORE's proposed construction. Indeed, this construction is consistent with the Court's conclusion that an empty payload field has zero data in it. In addition, the Court believes this interpretation is consistent with the description of the invention contained in the "Summary of the Invention" section of the specification of the '306 Patent. In the Court's view accepting Bellcore's proposed construction of this phrase would read the word "empty" out of the claim language. In Bellcore's own words its construction

would allow "any packet to be written into any frame." (D.I. 142 at 29). This construction is at odds with the claim's express language which requires the data to be written into "any *available empty payload field* of any of said frames." Accordingly, the Court concludes that the phrase "data in packetized format from any of said sources is written into any available empty payload field of any of said frames" means that packets are only put in frames which are empty, i.e. which have zero data in their payloads.

f. for transmitting data from each of said sources at its own desired bit rate via said bit stream and for transmitting data from said plurality of sources simultaneously via said bit stream

[12] Bellcore contends that this phrase means that the original bit rate of the sources is maintained for transmission of the source data in a bit stream emanating from a single transmitting device by multiplexing the packetized data within the transmitting device into frames as necessary. According to Bellcore, this means that the number of packets per second generated by a given source will equal the number of packets per second inserted into the output stream for that source. Bellcore relies on the specification of the '306 Patent for its construction, particularly emphasizing Figure 4. According to Bellcore, data from a source enters each interface 50 on data line 21. The data is packetized and stored in FIFO memory 57. The FIFO memories present the packets to the framers 53 in the order and at the rate that the packets are generated. Bellcore contends that the source data is transmitted simultaneously because it is "multiplexed," meaning the packets are interspersed into a common stream. In explaining its construction, Bellcore reiterates that the claimed generating and filling steps are not limited to any particular sequential order. (D.I. 142 at 30).

In response to Bellcore's argument, FORE contends that Bellcore's proposed definition is wrong, because it does not follow that "the source data is transmitted simultaneously because it is 'multiplexed.' " (D.I. 153 at 9-10). FORE also contends that the '306 Patent describes several "multiplexors" which are different that the DTDM assembler of Figure 4, which is the subject of Claim 1 of the '306 Patent. Thus, FORE contends that the use of the word "multiplexed" would be confusing in this context.

In proposing an alternate construction of this phrase, FORE contends that this language means that "two or more empty frames are filled at the same time by different data sources." (D.I. 146 at 15; D.I. 153 at 9-10). According to FORE, the interface unit of each source has access to the bit stream at the same time, so that the interface unit of each source can insert packets into passing empty frames at the same time. According to FORE, simultaneous transmission from multiple sources requires that each source have its own insertion point into the bits stream, otherwise the sources would not be transmitting simultaneously.

The Court agrees with FORE's construction of this phrase. In the Court's view, this construction is supported by both the plain meaning of the word "simultaneously" as used in the claim and by the specification of the '306 Patent. ('306 Patent, col. 13, lines 49-51). Accordingly, the Court construes the phrase "for transmitting data from each of said sources at its own desired bit rate via said bit stream and for transmitting data from said plurality of sources simultaneously via said bit stream" to require two or more empty frames to be filled at the same time by different data sources.

2. *Claim 3 of* the '306 Patent

In full, Claim 3 of the '306 Patent provides:

A method for generating a bit stream capable of transporting data originating from both circuit transmission and packet sources comprising

generating a bit stream comprising a sequence of frames, each of said frames including a transmission overhead field containing frame timing information and an empty payload field,

packetizing data from a plurality of sources having different bit rates and which have access to said bit stream including circuit transmission sources or customer premises equipment to produce data packets, and

inserting said packets from said sources into the empty payload fields of frames such that a packet from any of said sources is inserted into any available empty payload field of any said frames for transmitting data from each of said sources at its own desired bit rate via said bit stream and for transmitting data from said plurality of sources simultaneously using said bit stream.

The Court will examine each of the disputed terms and phrases below.

a. generating a bit stream

For the reasons discussed previously, the Court concludes that this phrase has the same meaning as described in the Court's analysis of this language in Claim 1.

b. frame timing information

For the reasons discussed previously, the Court concludes that this phrase also has the same meaning as described in the Court's analysis of this language in Claim 1.

c. an empty payload field

For the reasons discussed previously, the Court concludes that this phrase has the same meaning as described in the Court's analysis of this language in Claim 1.

d. inserting said packets from said sources into the empty payload fields of said frames

The parties' dispute concerning this language is essentially the same dispute raised in the context of the "filling the empty payload fields" language of Claim 1. Consistent with its argument related to the "filling" language in Claim 1, Bellcore contends that "inserting said packets from said sources into the empty payload fields of said frames" refers to outputting source data packets when available during a payload field interval. Consistent with its previous argument, FORE contends that the inserting step cannot occur until the empty frames have been generated. For the reasons discussed previously, the Court agrees with FORE and construes this language in the same fashion as the "filling" language in Claim 1.

e. such that a packet from any of said sources is inserted into any available empty payload field of any of said frames

For the reasons discussed previously, the Court construes this phrase in accordance with the construction provided by the Court in its analysis of the similar language FN3 used in Claim 1.

FN3. The language used in Claim 1 is virtually identical to this language except that Claim 1 reads "data in packetized format" instead of "a packet" and "written into" instead of "inserted into."

f. for transmitting data from each of said sources at its own desired bit rate via said bit stream and for transmitting data from said plurality of sources simultaneously using said bit stream

The parties' arguments regarding the construction of this phrase are premised on their arguments relating to the comparable language in Claim 1. Accordingly, for the reasons discussed previously, the Court will construe this phrase in accordance with the construction provided by the Court in its analysis of the comparable language used in Claim 1.

3. ***Claim 4 of*** the '306 Patent

In full, Claim 4 of the '306 Patent provides:

An apparatus for assembling a dynamic time division multiplexing bit stream comprising,

generating means for generating a train of frames wherein each frame includes a transmission overhead field containing timing information and an empty payload field,

processing means for processing data from a plurality of sources into packet format, and

inserting means for receiving said train of frames and for inserting each of said packets comprised of data from one of said plurality of sources into any empty payload field of any of said frames available to said inserting means to form said bit stream so that data from each of said sources can be transmitted at its own desired bit rate via said bit stream and so that data from said plurality of sources can be transmitted simultaneously via said bit stream.

[13] Although Claim 4 contains many of the same limitations included in Claim 1, the parties agree that Claim 4 is an apparatus claim which recites many limitations in a "means-plus-function" format. Means-plus function elements must be interpreted under 35 U.S.C. s. 112, para. 6. In pertinent part, Section 112, para. 6 provides:

An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claims shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereto.

Although use of means-plus-function language in a claim is permissible, a means clause does not encompass every means for performing the specified function. *The* Laitram Corporation v. Rexnord, 939 F.2d 1533, 1535 (Fed.Cir.1991). Rather, the limitation must be construed "to cover the corresponding structure, material, or acts described in the specification and equivalents thereof." Odetics, Inc. v. Storage Technology Corp., 185 F.3d 1259, 1999 WL 455530, (Fed.Cir. July 6, 1999).

With these principles in mind, the Court will examine each of the disputed terms and phrases of Claim 4.

a. bit stream

In disputing the meaning of this term, Bellcore requests construction of more than the phrase "bit stream." Bellcore requests construction of the phrase "dynamic time division multiplexing bit stream." According to Bellcore, this phrase refers to a desired output transmission stream of bits that includes a sequence of

transmission frames and dynamically multiplexed packets, i.e. packets multiplexed into the assembled bit stream at the rate they are generated.

FORE does not proffer a definition for the entire phrase "dynamic time division multiplexing bit stream," but merely contends that bit stream means the same as its proffered definition for the phrase "generating a bit stream" in Claim 1.

For the reasons discussed previously, the Court rejects FORE's definition of a bit stream as being limited to a serial configuration of bits. However, because FORE does not address Bellcore's argument or the language "dynamic time division multiplexing," it is unclear to the Court whether FORE agrees or disputes Bellcore's construction of this term. Accordingly, the Court will reserve decision on the meaning of this phrase until such time as the parties' clarify their respective positions.

b. generating means for generating a train of frames wherein each frame includes a transmission overhead field containing frame timing information and an empty payload field

[14] The parties agree that this claim element is recited as a means-plus-function element. Bellcore contends that the disclosed structure corresponding to the "generating means" includes control 210, tristate device 222, ROM 224 and timing generator 209. Bellcore contends that these are the only structural elements shown in the specification which are involved in performing the claimed function of generating empty frames, and therefore Bellcore contends that it is improper to identify the entire framer as the structure corresponding to the generating means.

FORE contends that the structure of the generating means is the entire framer unit shown in Figure 12 and described in the specification. ('306 Patent, col. 16, line 8-col. 17, line 30). In response to Bellcore's reliance on only a portion of the framer unit, FORE contends that Bellcore provides "no analysis that some subset of the structure could even work without all of the components on that page." (D.I. 153 at 11). To this effect, FORE points out that the specification acknowledges the framer unit as an "important component" of the process. ('306 Patent, col. 15, lines 8-11).

The Court agrees with Bellcore. Although the specification repeatedly explains that the framer unit generates trains of empty frames, the specification also expressly identifies those structures of the framer unit involved in the generating means as control 210, tristate device 222, ROM 224 and timing generator 209. ('306 Patent, col. 16, lines 27-31; col 16, lines 62-col. 17, line 7). That the entire framer unit is not involved in the process is further confirmed by the specification which expressly provides that "the serial input 202 and *associated serial-to-parallel converter 212 are not utilized*." ('306 Patent, col. 16, lines 64-65 (emphasis added)). Accordingly, the Court concludes that the structures corresponding to the "generating means" are control 210, tristate device 222, ROM 224 and timing generator 209.

c. processing means for processing data from a plurality of sources into packet format

[15] The parties' agree that this element is asserted in a means-plus-function format. However, Bellcore contends that the disclosed structure corresponding to this language includes packetizers 55 and their equivalents. Bellcore contends that this does not include the FIFOs because the FIFOs store packets after they have been formed and are not involved in packetizing.

FORE contends that the disclosed structures corresponding to this language are both the packetizers 55 and the FIFOs. Although FORE contends that it is essential for each source to have its own packetizer and FIFO,

FORE's argument focuses on the reason for including the packetizer and not on its reason for including the FIFO.

The Court agrees with Bellcore that the structures indicated in the specification for performing this function are the packetizers 55 and not the FIFO memories. As the specification states, "packetizer 55 puts the incoming data into a packet structure" and it is only " *after* the data is put into a packet structure" by the packetizers 55, that the data "is stored in a FIFO 57." ('306 Patent, col. 9, lines 19-21, 28-30 (emphasis added)). Accordingly, the Court concludes that the packetizers 55 are the structures which perform the function recited in this element.

d. inserting means for receiving said train of frames and for inserting each of said packets comprised of data from one of said plurality of sources into any empty payload field of any of said frames available to said inserting means to form said bit stream so that data from each of said sources can be transmitted at its own desired bit rate via said bit stream and so that data from said plurality of sources can be transmitted simultaneously via said bit stream

[16] As with the prior elements, the parties agree that this element is recited in a means-plus-function format. Bellcore contends that the structures corresponding to the "inserting means" include control 210, tristate device 218, tristate device 220, frame detect 214 and timing generator 209.

In contrast to Bellcore's designations, FORE contends that the inserting means is multiple framer units arranged in a daisy chain. FORE directs the Court to Figure 4 which depicts a daisy chained configuration of framer units inserting packets into an empty train of frames and to Figure 12 and its accompanying text which describes how the framer 200 writes packet data into empty frames. ('306 Patent, col. 16, lines 32-61).

The Court agrees with Bellcore. The specification of the '306 Patent explains the inserting means in detail. ('306 Patent, col. 16, lines 39-58). According to the specification, the timing information for the framer unit is provided by timing generator 209. The DTDM frame is converted to parallel form and is detected by frame detector 214. The frame detector 214 is in communication with the control 210 which detects whether the frame is empty. The frame cannot reach the parallel to serial converted unless the control 210 applies a signal to the tristate device 218. If the frame is empty, a signal is applied by the control 210 to the tristate device 220 to enable the data to be inserted into the particular frame. ('306 Patent, col. 16, lines 43-58).

With regard to FORE's position that the structural means is multiple framer units arranged in a daisy chain, the Court acknowledges that the specification indicates that the "framer units *may* be connected in a daisy chain fashion." ('306 Patent, col. 16, lines 32-33 (emphasis added)). However, a daisy chain configuration is not a mandatory requirement. Further, as indicated above, the specification precisely outlines those structures which are involved in the insertion means. Accordingly, the Court concludes that the structures corresponding to the "inserting means" are control 210, tristate device 218, tristate device 220, frame detect 214 and timing generator 209.

B. *The* '768 Patent

Bellcore asserts Claims 13 of the '768 Patent. In full, Claim 13 reads:

The method for demultiplexing a serial data bit stream consisting of a continuum of an interleaved multiplicity of data bytes of predetermined size derived from a plurality of identically-formatted

contributory frames each containing a plurality of said data bytes, and for reconstructing said data bytes and identifying from among them a benchmark from which may be determined the beginning byte of each of such contributory frames and, thereby, the boundaries of such frames, said method comprising:

(a) accumulating data bits from said serial stream to form bytes having the same predetermined number of bits as do said interleaved data bytes;

(b) comparing at least one bit pattern from each byte thus formed with at least one bit pattern known to have comprised a byte of each of said contributory frames;

(c) providing a first signal when a match is detected between said compared patterns;

(d) effecting in response to said first signal the output of the byte of matching bit pattern, and each byte thereafter formed of newly accumulated bits, as said reconstructed bytes;

(e) comparing bit patterns from a contiguous plurality of said output reconstructed bytes with bit patterns known to have comprised a like contiguous plurality of bytes of each of said contributory frames;

(f) providing a second signal as said benchmark identification when a match is detected between said compared patterns; and

(g) effecting in response to said second signal discontinuation of the provision of said first signal.

The Court will examine each of the disputed terms and phrases below.

1. demultiplexing a serial data bit stream consisting of a continuum of an interleaved multiplicity of data bytes of predetermined size derived from a plurality of identically-formatted contributory frames each containing a plurality of said data bytes

[17] According to Bellcore, the phrase "demultiplexing a serial data bit stream" means detecting the bits in the serial bit stream and converting them to a sequence of byte groupings. (D.I. 142 at 38; D.I. 144 at 3). Bellcore contends that the term "demultiplexing" is not defined in the '768 Patent, but that one skilled in the art would understand that bit streams are "demultiplexed" or converted to a sequence of data bytes representing the bytes of the original frame. Relying on the specification, Bellcore explains that "[i]n the general application of the present invention, the high-speed serial bit stream of the STS-N level, e.g. STS-24 transmission is demultiplexed to the basic SONET 8-bit byte-parallel format..." ('768 Patent, col. 3, lines 8-11). The SONET frames "are transmitted in the continuing serial bit stream to their destined terminating SONET receiver where the frames must be reformatted by reconstructing and demultiplexing the transmitted bytes in order and sequence." ('768 Patent, col. 4, lines 22-27). According to Bellcore, the specification further explains that the invention "provides for the maximum utilization of available technologies for optimum economies of power and time in the demultiplexing of high speed serial bit data transmission to low-speed byte-parallel format within the SONET signal hierarchy." ('768 Patent, col. 9, lines 49-54).

Bellcore further defines the phrase "consisting of a continuum of an interleaved multiplicity of data bytes of predetermined size derived from a plurality of identically-formatted contributory frames each containing a plurality of said data bytes" to mean that the serial bit stream consists of transmitted SONET-like frames greater than STS-1. In proffering this definition, Bellcore specifically defines "a continuum of an

interleaved multiplicity of data bytes of predetermined size" as a sequence of interspersed bytes, usually eight bits in size, but not required to be eight bits in size. Bellcore also defines "derived from a plurality of identically-formatted contributory frames each containing a plurality of said data bytes" as the format of the sequence of interspersed bytes. Specifically, Bellcore contends that this phrase means having a format based on multiple basic frames having the same byte format.

In contrast to Bellcore's interpretation, FORE contends that the preamble requires each frame of the serial bit stream that is being demultiplexed to have been formed by interleaving all eight bit bytes of two or more contributory frames. (D.I. 146 at 25). In other words, FORE's definition focuses on how the serial bit stream referred to in the claim was formed. According to FORE, the serial bit stream consists of a train of repeating higher-level frames. Each higher level frame is constructed by combining two or more pre-existing, distinct and complete lower-level frames called "contributory frames." FORE further contends that this language means that there can be no gaps or pauses or breaks in the interleaving of the contributory frames. In support of its position, FORE contends that the specification makes clear that demultiplexing is the process of disassembling previously multiplexed (or combined) lower-level SONET frames. ('768 Patent, col. 1, lines 37-44; col. 2, lines 28-30; col. 5, lines 41-47). Stated another way, FORE contends that "demultiplexing" is the process of separating higher-level, higher speed STS-N frames transmitted in the serial data bit stream into the lower-level, lower speed STS-1 frames that make up each STS-N frame. Thus, according to FORE, the claim is limited to taking multiplexed STS-N frames and separating them into the original STS-1 frames that were previously combined to create the STS-N frames.

In addition, FORE specifically disputes Bellcore's construction of the words "continuum," "interleaved" and "byte." FORE contends that the word "continuum" means "continuous" and not "a sequence" as Bellcore contends. As for Bellcore's interpretation of the words "interleaved" and "byte," FORE contends that Bellcore's definitions are overly-broad, and therefore, inconsistent with the patent's specification.

The Court agrees with FORE that the preamble limits this claim of the patent to taking multiplexed STS-N frames and separating them into the original STS-1 frames that were previously combined to create the STS-N frames. The specification of the '768 Patent speaks solely in terms of multiple, lower-speed frames that are multiplexed together to form a higher speed frame. The patent does not address the type of SONET framing structure which consists of larger frame formats generated in a single operation, using the higher bit rate and multiple STS-1 frame format. In order for the invention to perform its demultiplexing function, the bit stream must be multiplexed. The specification of the '768 patent expressly defines a multiplexed bit stream as follows: "[A] multiplexed serial bit stream is assembled by interleaving repeated sequential extractions of one byte from each of the component STS-1 frames." ('768 Patent, col. 1, lines 37-40). The patent specification further provides:

It is *necessary,* therefore, that the signal receiver [the device receiving the SONET-formatted serial bit stream] *reconstruct* from this serial bit stream *the original base frame,* or some frame multiple thereof, in order that the correct substance of the transmitted signal may be recovered.

* * * * * *

[The] stream must be reformatted into the *original bytes and frames* in order for the receiver processing circuitry to property extract the transmitted data and messages.

('768 Patent, col. 1, lines 37-44 (emphasis added); col. 2, lines 28-30 (emphasis added)). Further, the

specification explains:

Upon completion of the formatting of the high-speed input serial data stream to a low-speed, properly synchronized byte-parallel data stream, there remains the problem of identifying the boundaries of *each frame of the original transmission* in order that the payload, as well as the relevant overhead information bytes, may be *demultiplexed to the basic STS-1 level*.

('768 Patent, col. 5, lines 41-47 (emphasis added)). Accordingly, based on the claim language and the specification, the Court concludes that the phrase "demultiplexing a bit stream" refers to taking multiplexed STS-N frames and separating them into the original STS-1 frames that were previously combined to create the STS-N frames.

[18] The Court further agrees with FORE that the serial bit stream that is being demultiplexed must have been formed by interleaving the bytes of two or more contributory frames, that each of the contributory frames must have exactly the same format, and that there can be no gaps or pauses in the interleaving. In the Court's view, the specification and Figure 2 support FORE's position that the bytes are continuously interleaved from each of the contributory frames. ('768 Patent, col. 1, lines 37-40; col. 4, lines 9-27). Indeed, byte-interleaving is described in the specification as requiring a byte to be taken, one at a time, from the same byte position of each contributory frame and placed in the bit stream. ('768 Patent, col. 4, lines 9-27). However, the Court cannot accept FORE's contention that a byte is limited to 8 bits. Although Bellcore concedes that a byte is usually 8 bits, the Court cannot conclude that the patent limits a byte to 8 bits, because the express language of the claim indicates that the data bytes are of a "predetermined size." If bytes were limited to 8 bits only, then the language "predetermined size" would be superfluous and unnecessary. Each and every word in a claim must have meaning and cannot be ignored. *See e.g.* Exxon Chem. Patents, Inc. v. Lubrizol Corp., 64 F.3d 1553, 1557 (Fed.Cir.1995). Accordingly, in order to give effect to the language "predetermined size" the Court will not exclude the possibility that a byte could be more than 8 bits.

As to the parties' dispute concerning the phrase "consisting of," the Court concludes that in this case, the phrase "consisting of" is used as a transition between the claim element "a serial data bit stream" and the clause that follows and modifies the claim element. Because the phrase "consisting of" introduces the modifying clause, "a continuum of an interleaved multiplicity of data bytes of a predetermined size derived from a plurality of identically-formatted contributory frames each containing a plurality of said data bytes," the Court concludes that it excludes any bit stream that does not have the exact limitations recited in the modifying clause. Georgia-Pacific Corp. v. U.S. Gypsum Co., 195 F.3d 1322, 1327 (Fed.Cir.1999); Mannesmann Demag Corp. v. Engineered Metal Prods. Co., 793 F.2d 1279, 1282 (Fed.Cir.1986).

In sum, the Court concludes that the phrase: "The method for demultiplexing a serial data bit stream consisting of a continuum of an interleaved multiplicity of data bytes of predetermined size derived from a plurality of identically-formatted contributory frames" means that each frame of the serial bit stream that is being demultiplexed was formed by interleaving the bytes of two or more contributory frames. Each contributory frame must have exactly the same format and there can be no gaps or pauses in the interleaving.

2. reconstructing said data bytes; said reconstructed data bytes

[19] In interpreting this phrase, FORE contends that "reconstructed bytes" are bytes from the contributory

frames that have been aligned, latched, and converted from serial to parallel format. Bellcore contends that the phrase means "identifying the byte groupings of the transmitted frame." (D.I. 151 at 14). However, Bellcore further contends that it essentially agrees with Fore's proposed construction, but that it is inappropriate to include the "latched" limitation in step (d) where the word "reconstructed bytes" appears.

The Court disagrees with Bellcore. The specification describes the process of reconstructing bytes as aligning, latching and converting the bytes from serial to parallel format. Detecting the first frame byte F1 by comparing bits in the incoming serial stream to the known bit pattern "trigger[s] the output of the matched F1 framing byte and begin[s] clocking *reconstructed, properly-phased 8 bit bytes out of latch 33* ..." Patent, col. 5, lines 31-36 (emphasis added). Thus, as the specification indicates, latching is part of the reconstruction process, and reconstruction is not simply identifying byte groupings, as Bellcore contends. Accordingly, the Court concludes that "reconstructed bytes" or "reconstructing said data bytes" refers to bytes that have been aligned, latched and converted from serial to parallel format.

3. identifying from among them a benchmark from which may be determined the beginning of each of such contributory frames and, thereby, the boundaries of such frames

[20] In disputing the meaning of this phrase, Bellcore contends that the term "benchmark" means marker and the term "boundaries" means the beginnings or ends of certain portions of a frame. Bellcore contends that the transition from F1 to F2 framing bytes serves as the "benchmark." (D.I. 252 at 15).

FORE contends that this limitation means that (1) a benchmark must be determined for each instance of higher-level frames, (2) from the benchmark the first byte of each distinct and complete higher-level frame must be determined, and (3) from that first byte, the boundary of each distinct and complete higher-level contributory frame must be determined. (D.I. 253 at 3-4). FORE does not appear to disagree with Bellcore's definition of "benchmark." Additionally, FORE agrees that the term "boundaries" means the beginning or end of a frame, but disagrees to the extent that Bellcore construes this term to mean "certain other portions" of a frame, which FORE contends is vague.

It appears to the Court that the parties central disagreement concerning the meaning of this phrase is whether the frame boundaries "must" be determined as FORE contends or "may" be determined as Bellcore contends. Reading the claim language as a whole in light of the specification, the Court agrees with FORE that this element requires the beginning of each frame, and therefore the frame boundaries, to be determined from the benchmark. As the specification states, "[I]t is essential to the proper recovery of the *original SONET frames* that the byte assembly be correctly synchronized and *the boundaries of each frame* be identified in the bit stream transmission in order that the reconstructed bytes will duplicate each of the bytes which were interleaved to produce that serial transmission signal." ('768 Patent, col. 1, lines 47-53 (emphasis added)). In the Court's view, accepting Bellcore's contention renders the requirement of identifying the benchmark a useless exercise. The very purpose of identifying the benchmark is to determine the "beginning byte" and the "boundary" of each contributory frame, and therefore, the Court cannot accept a construction of the phrase "may be" which would make this function optional. Accordingly, the Court concludes that this phrase requires that (1) a benchmark must be determined for each instance of higher-level frames, (2) from the benchmark the first byte of each distinct and complete higher-level frame must be determined, and (3) from that first byte, the boundary of each distinctand complete higher-level contributory frame must be determined.

4. comparing bit patterns from a contiguous plurality of said output reconstructed bytes with bit

patterns known to have comprised a like plurality of bytes of each of said contributory frames

[21] FORE contends that this phrase requires "bit patterns from two or more *adjacent* reconstructed bytes to be compared with bit patterns known to have been present in two or more *adjacent* bytes in *each* of the contributory frames which have been multiplexed to form the bit stream." (D.I. 146 at 31 (emphasis in original)). FORE further contends that each contributory frame must both contain the pattern and contain it within two or more adjacent bytes of the contributory frame.

In response to FORE's construction, Bellcore contends that this phrase means "that the contributory frames each have a plurality of bytes like the bytes included in the compared 'contiguous plurality' of bytes." (D.I. 151 at 16). According to Bellcore, this phrase cannot mean, as FORE contends, that the compared contiguous plurality must exist within each contributory frame. According to Bellcore, the requirement for byte interleaving makes this arrangement impossible.

The Court agrees with FORE's construction of this phrase. This element of the patent concerns frame boundary detection *after* byte reconstruction, because as the claim recites, the comparison is made from a "contiguous plurality of output *reconstructed* bytes." ('768 Patent, col. 12, line 12). Because this step is performed after the interleaved bytes are reconstructed, Bellcore's contention that the interleaving arrangement makes this interpretation impossible is incorrect. As the specification states:

Upon completion of the formatting of the high-speed serial data stream to a low-speed, *properly synchronized* byte-parallel data stream, there remains the problem of identifying the *boundaries of each frame* of the original transmission in order that the payload, as well as the relevant overhead information bytes, may be demultiplexed to the basis STS-1 level. For this purpose, the present invention relies upon the prescribed bit sequence of both the F1 and F2 framing bytes, of which each frame above STS-1 will have at least two, F2 bytes following immediately upon the final F1 framing byte as depicted in FIG. 2.

('768 Patent, col. 5, lines 41-52 (emphasis added)). The reference to Figure 2 here is only for the purpose of illustrating that the F2 bytes follow the F1 bytes. Accordingly, the Court concludes that "comparing bit patterns from a contiguous plurality of said output reconstructed bytes with bit patterns known to have comprised a like contiguous plurality of bytes of each of said contributory frames" means that two or more adjacent reconstructed bytes are compared with bit patterns known to have been present in two or more adjacent bytes in each of the contributory frames which have been multiplexed to form the bit stream. Each contributory frame must contain the bit pattern and each frame must contain the bit pattern within two or more adjacent bytes of the contributory frame.

CONCLUSION

For the reasons discussed, the Court has construed the disputed terms of the '306 and '768 as provided herein. An Order consistent with this Opinion will be entered setting forth the meaning of the disputed terms in the '306 and '768 Patents.

ATTACHMENT

ATTACHMENT

United States Patent [19]

Chao et al.

[11] Patent Number: 4,893,306

[45] Date of Patent: Jan. 9, 1990

[54] METHOD AND APPARATUS FOR MULTIPLEXING CIRCUIT AND PACKET TRAFFIC

[75] Inventors: Hung-Hsiang J. Chao, Madison; Sang H. Lee, Bridgewater; Liang T. Wu, Gladstone, all of N.J.

[73] Assignee: Bell Communications Research, Inc., Livingston, N.J.

[21] Appl. No.: 118,977

[22] Filed: Nov. 10, 1987

[51] Int. Cl.[4] H04J 3/16; H04J 3/26

[52] U.S. Cl. 370/94.3; 370/84; 370/99; 370/112

[58] Field of Search 370/94, 60, 84, 99, 370/111, 112, 82, 110.1, 89

[56] References Cited

U.S. PATENT DOCUMENTS

4,321,703	3/1982	Schwiderski et al.	370/89
4,516,240	5/1985	Kume et al.	370/94
4,594,708	6/1986	Servel et al.	370/94
4,685,103	8/1987	Shikama et al.	370/89
4,706,246	11/1987	Kume	370/89
4,763,319	8/1988	Rotenbilt	370/89
4,764,921	8/1988	Graves et al.	370/110.1
4,771,425	9/1988	Baran et al.	370/110.1

OTHER PUBLICATIONS

R. W. Mulse, et al., "Experiments in Wideband Packet Technology", Proc. 1986, International Zurich Seminar on Digital Communications, pp. 126-138.

W. W. Chu, "A Study of Asynchronous Time Division Multiplexing for Time Sharing Computer Systems", Proc. AFIPS, vol. 35, pp. 669-678, 1969.

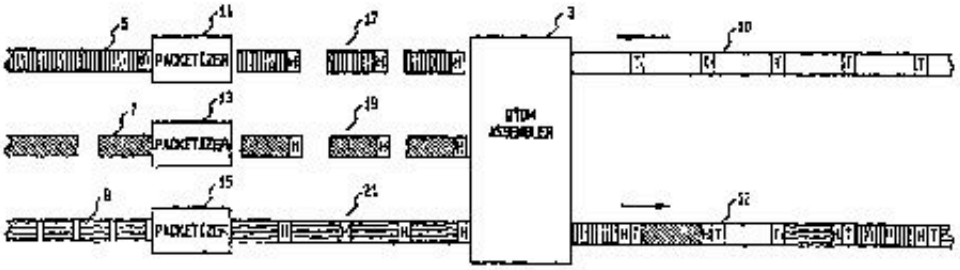
A. Thomas, et al., "Asynchronous Time Division Techniques: An Experimental Packet Network Integrating Video Communication", Proc. International Switching Symposium, May 1984.

Primary Examiner—Douglas W. Olms
Assistant Examiner—Min Jung
Attorney, Agent, or Firm—James W. Falk

[57] ABSTRACT

A data transmission technique referred to herein as Dynamic Time Division Multiplexing (DTDM) is disclosed along with a set of multiplexers and demultiplexers required to apply DTDM in an actual telecommunications network. The DTDM technique uses a transmission format which is compatible with the existing digital circuit transmission format and the packet transmission format so that DTDM is able to handle the transmission of circuit and packet traffic. Thus, DTDM provides a flexible migration strategy between present circuit networks and future broadband packet networks.

7 Claims, 10 Drawing Sheets



*657

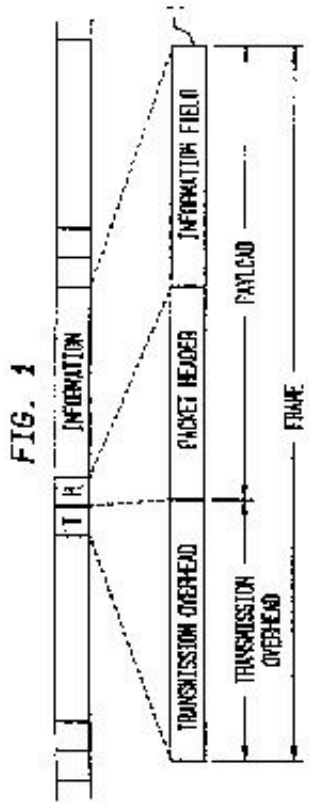


FIG. 1

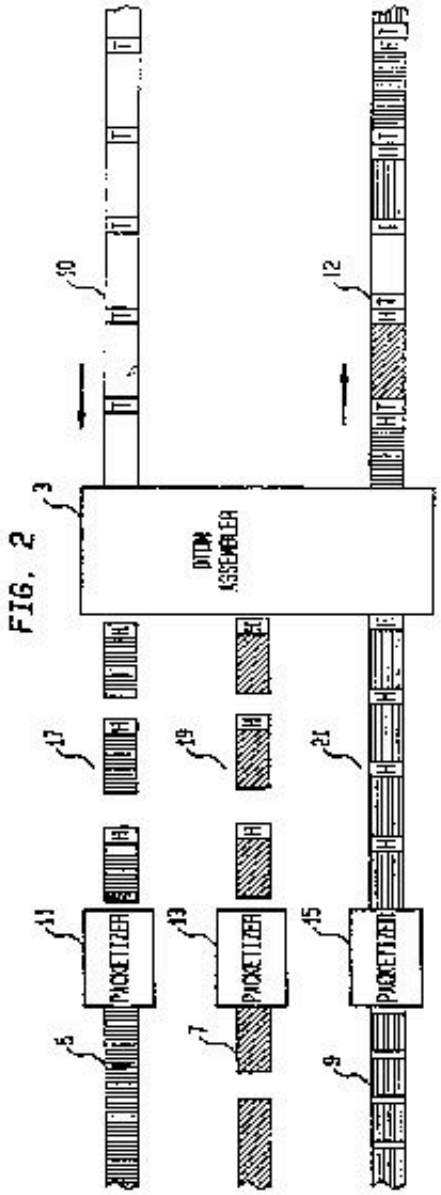


FIG. 2

FIG. 3

FIG. 3

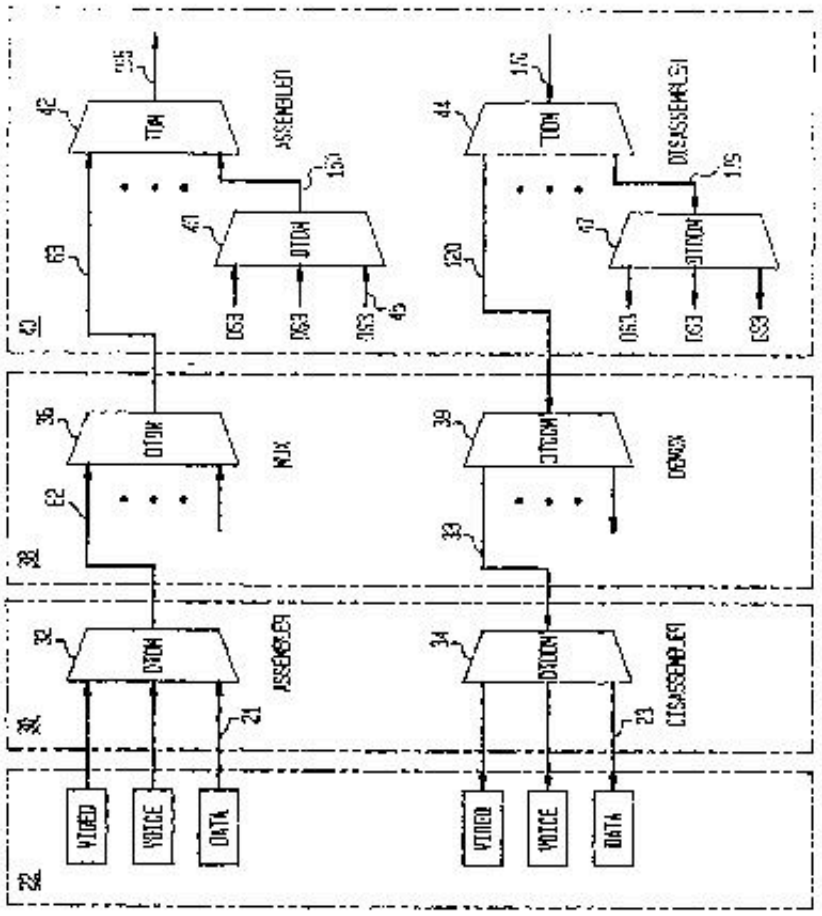

30
42
TDM
ASSEMBLER
155
47
DTDM
63
DS3
DS3
DS3
45
40
44
TDM
DISASSEMBLER
175
47
DTDM
120
DS3
DS3
DS3
36
DTDM
MUX
62
38
39
DTDM
DEMUX
39
32
DTDM
ASSEMBLER
21
30
34
DTDM
DISASSEMBLER
23
VIDEO
VOICE
DATA
VIDEO
VOICE
DATA
22

FIG. 4

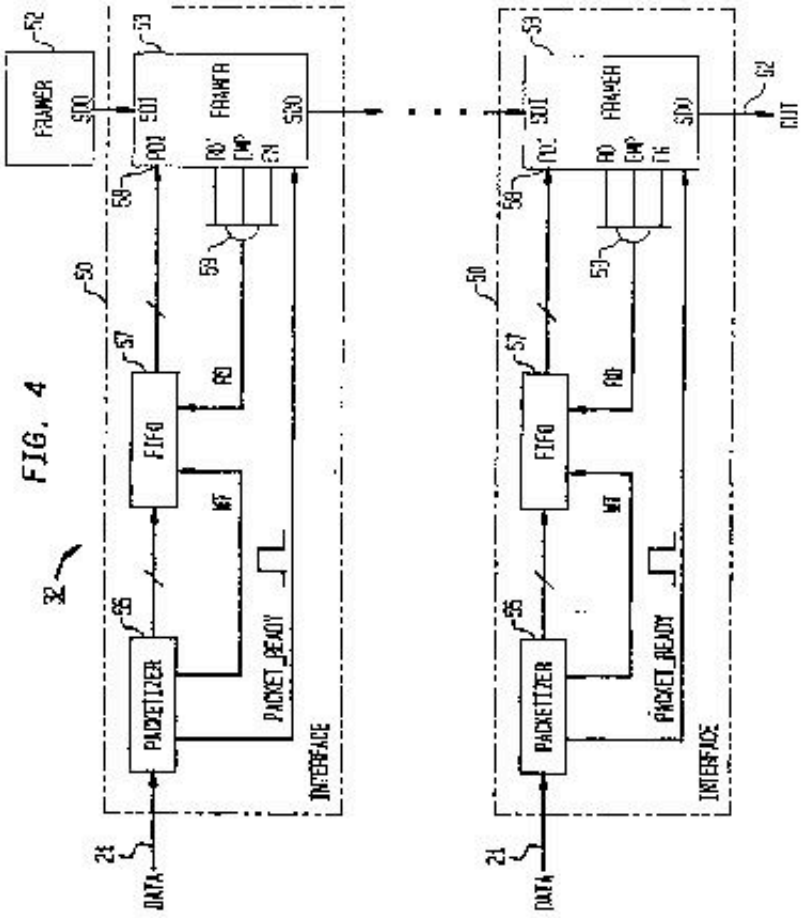

FRAMER
FRAMER
FIFO
PACKETIZER
PACKET_READY
INTERFACE
DATA
OUT

FIG. 4

FIG. 5

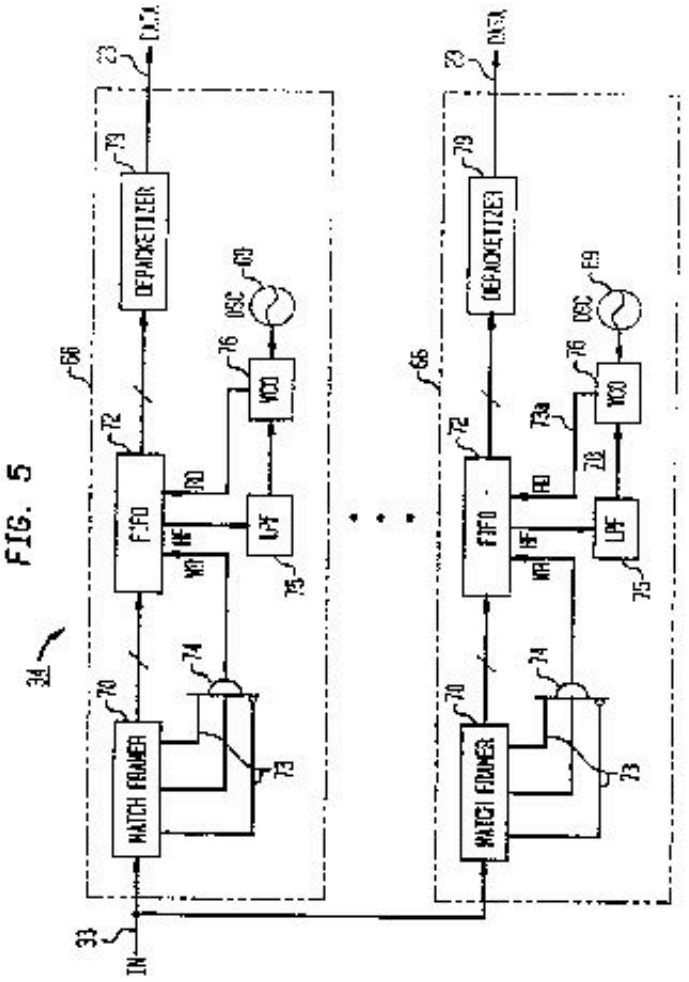


FIG. 5
34
33
IN
DATA
DEPACKETIZER
OSC
VCO
FIFO
LPF
MATCH FRAMER

FIG. 6

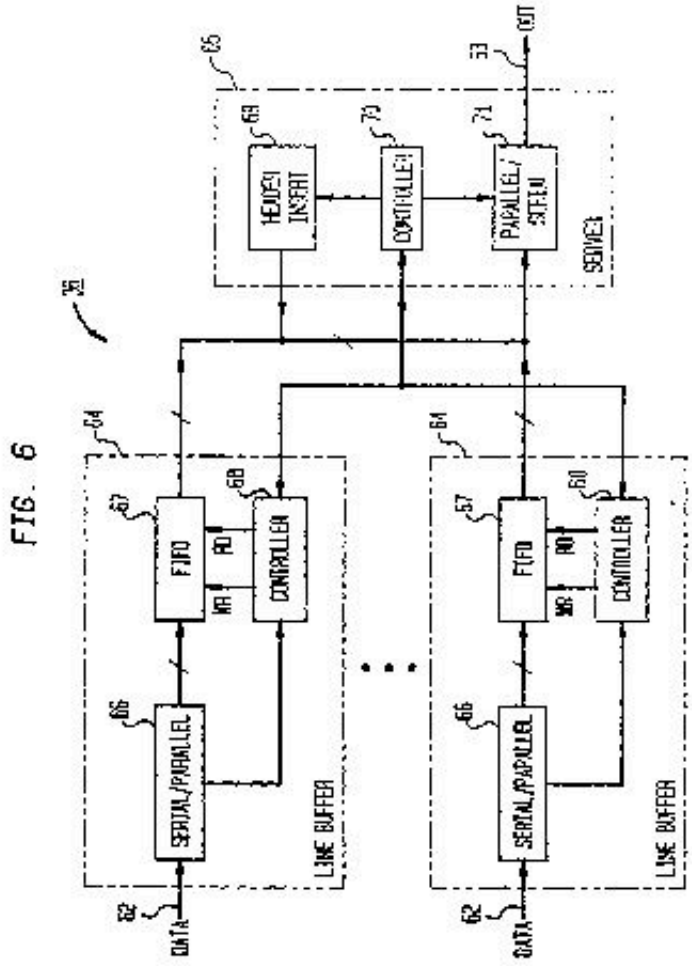

65
33
OUT
68
70
71
HEADER INSERT
CONTROLLER
PARALLEL/SERIAL
SERVER
75
64
61
67
68
FIFO
RD
WR
CONTROLLER
66
SERIAL/PARALLEL
LINE BUFFER
62
DATA

FIG. 6

FIG. 7

FIG. 7

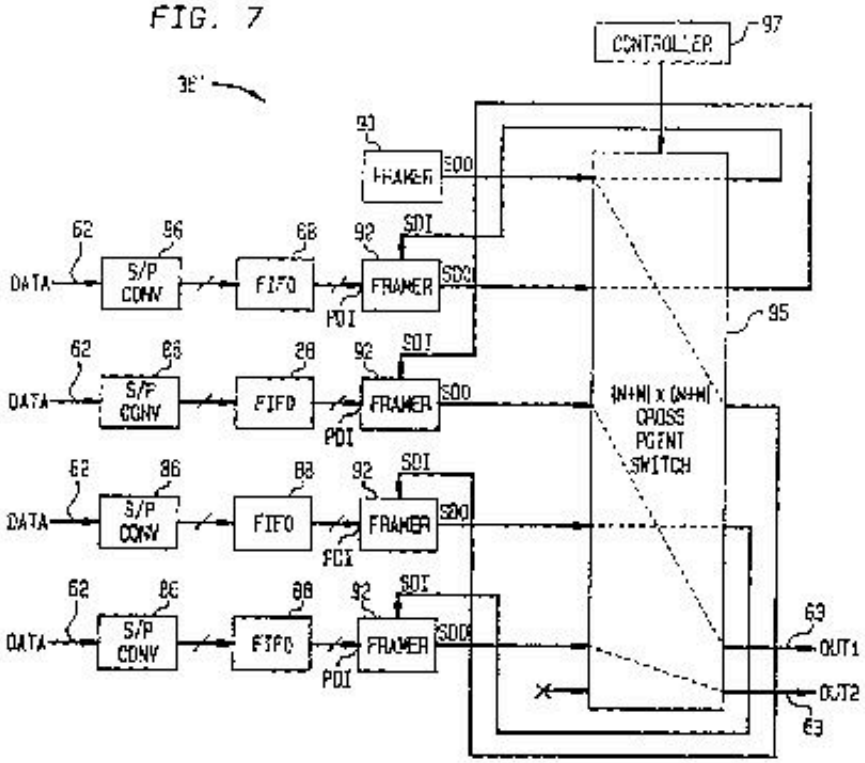


FIG. 8

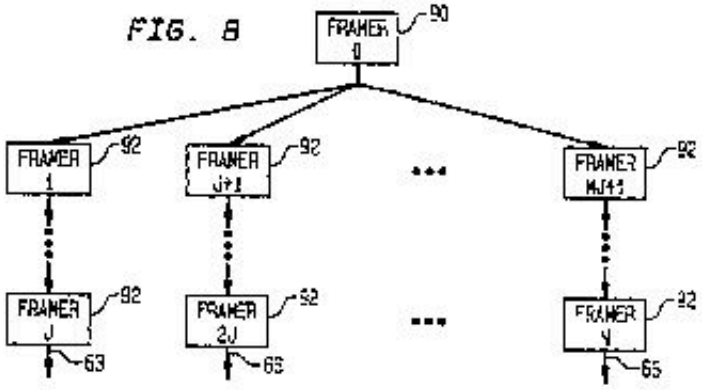


FIG. 8

FIG. 9

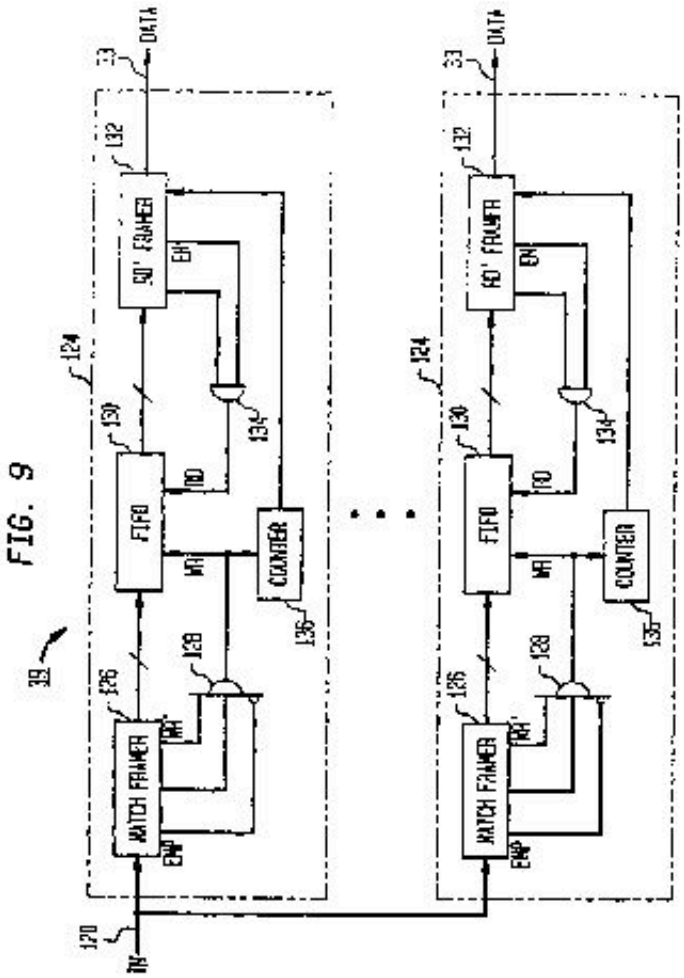

DATA
33
132
SOF FRAMER
EN
124
130
FIFO
RD
WR
134
COUNTER
136
128
126
MATCH FRAMER
RST
EMP
170
IN
39
EOF FRAMER

FIG. 9

FIG. 10

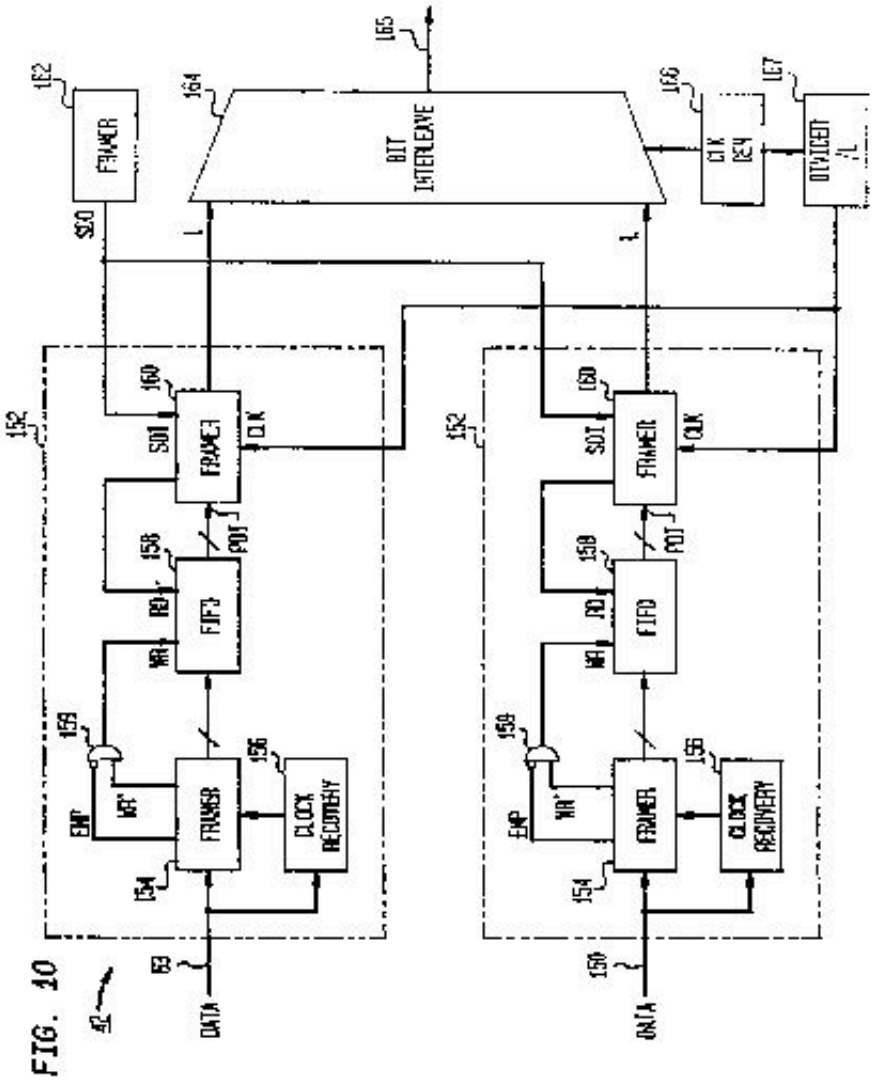

FRAMER
BIT INTERLEAVE
CLK DIVIDER
FRAMER
FIFO
FRAMER
CLOCK RECOVERY
DATA

FIG. 10

FIG. 11

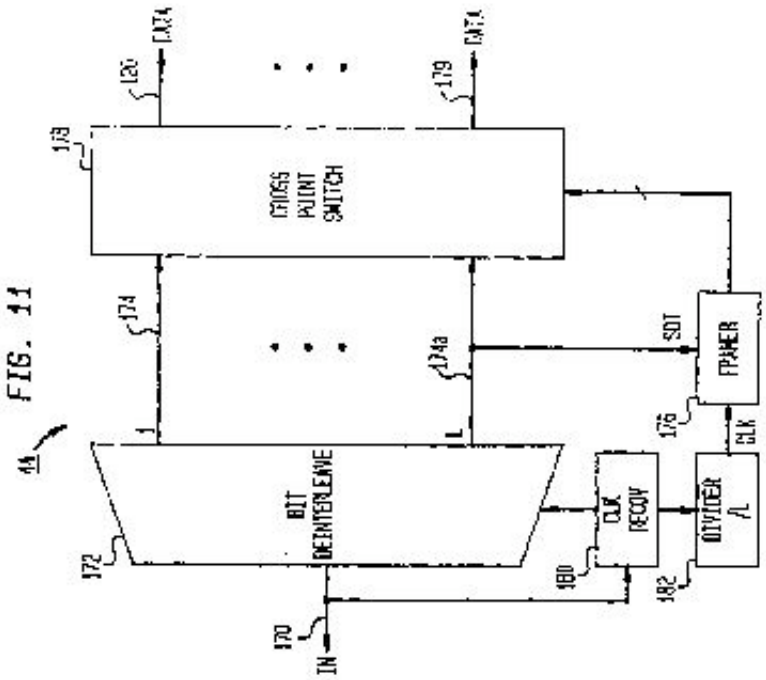


FIG. 11
44
DATA
120
DATA
179
168
CROSS POINT SWITCH
174
174a
SDT
FRAMER
176
CLK
BIT DEINTERLEAVE
CLK RECOV
DIVIDER /L
172
180
182
170
IN

FIG. 12

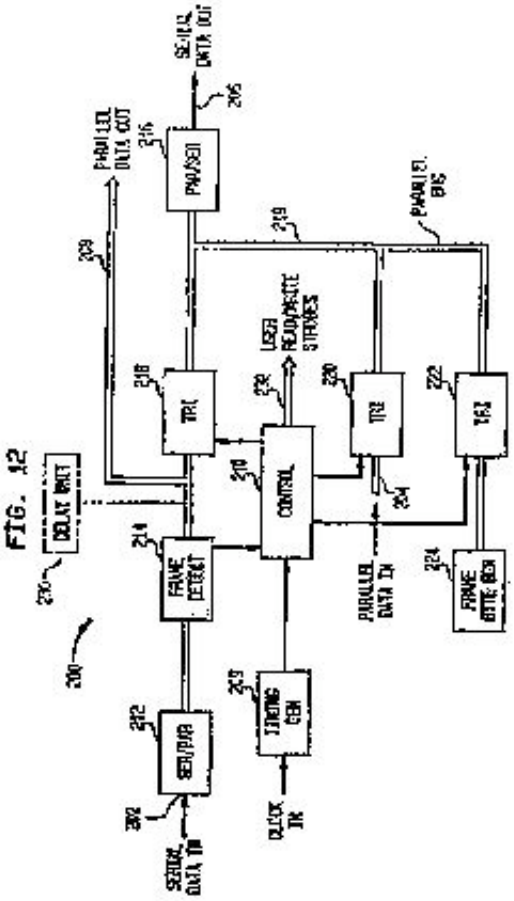


FIG. 12
DELAY UNIT
FRAME DETECT
CONTROL
TIMING GEN
SER/PAR
PAR/SER
TR1
TR2
TR3
FRAME BYTE GEN
SERIAL DATA OUT
PARALLEL DATA OUT
PARALLEL BUS
LASER MODULATE STROBES
PARALLEL DATA IN
SERIAL DATA IN
CLOCK IN
200
202
204
206
208
210
212
214
216
218
219
220
222
224
230

FIG. 13

METHOD AND APPARATUS FOR MULTIPLEXING CIRCUIT AND PACKET TRAFFIC

RELATED APPLICATIONS

The following applications contain subject matter related to the subject matter of the present application, are assigned to the assignee hereof and have been filed on the same date as the present application.

1. J. J. Chao, "DTDM Multiplexer With Cross-Point Switch", Ser. No. 118,979, now U.S. Pat. No. 4,855,999, issued Aug. 8, 1989
2. M. W. Beckner, F. D. Porter, K. Shu, "DTDM Multiplexing Circuitry", Ser. No. 118,897, now U.S. Pat. No. 4,833,671, issued May 23, 1989
3. H. J. Chao, S. H. Lee, "Time Division Multiplexer for DTDM Bit Streams", Ser. No. 118,978, now U.S. Pat. No. 4,833,673, issued May 23, 1989
4. M. W. Beckner, J. J. Chao, T. J. Robe, L. S. Smoot, "Framer Circuit", Ser. No. 118,898, now U.S. Pat. No. 4,819,226, issued Apr. 4, 1989.

FIELD OF THE INVENTION

This invention relates to the transmission of data in telecommunications networks. More particularly, the present invention relates to a data transmission technique referred to herein as Dynamic Time Division Multiplexing (DTDM), and a set of multiplexers and demultiplexers required to apply DTDM in an actual telecommunications network. DTDM is capable of effectively handling both circuit and packet traffic and thus provides a migration strategy between the present circuit switched telephone network and the future broadband packet switched network.

BACKGROUND OF THE INVENTION

Presently, there are significant uncertainties when it comes to predicting the future demand for broadband telecommunications services such as high definition video and interactive data communications. This uncertainty in the future demand for broadband telecommunications services has a significant impact on the design of public telephone networks. First, to satisfy the unknown growth pattern in future service demands, it is desirable to have a robust network design that can be easily modified in response to changes in demand for particular telecommunications services. Second, the network must be able to handle vastly different types of traffic ranging from low speed data and voice to full motion video. Third, depending on the demand for wideband services, a network design must be capable of providing a migration strategy from existing copper wires and circuit transmission and switching facilities to optical fibers and the succeeding generations of high speed packet transmission and switching facilities, which packet facilities are used in connection with the delivery of wideband telecommunications services. These three criteria determine the selection of the three major components of a network design: network topology, transmission systems and switching systems. Here, the concern is primarily with transmission systems and transmission techniques which meet the foregoing criteria.

Two important types of commercially used transmission systems are circuit systems and packet systems. Typically, circuit systems utilize time division multiplexing (TDM) as a transmission technique. When TDM is used, each data stream comprises frames which are subdivided into slots. Corresponding slots in each frame are allocated to specific connections. For example, the first slot in each frame is allocated to one specific connection and the second slot in each frame is allocated to a second connection, etc. Each frame also includes a field which contains transmission overhead information including frame synchronization words and control words. This traditional circuit transmission format can be extended to multiple bit rate services by allocating multiple slots in each frame to high bandwidth services. In such circuit transmission systems, a combination of space division switching and time division switching is utilized at the network switches to swap time slots between various bit streams so that connections to and between specific subscribers are established.

Historically, the first digital circuit transmission systems were introduced during the 1960's. These first digital circuit transmission systems were introduced in inter-office trunking applications to carry 24 voice channels by a single 1.544 Mb/sec digital stream. This is known as the DS-1 signal. Subsequently, the wide deployment of digital channel banks in the public telephone network required the multiplexing of several DS-1 signals into a higher speed bit stream to efficiently utilize available transmission links. As the network grew further, continuing efforts to effectively multiplex tributaries having different bit rates into a common bit stream resulted in the well-known hierarchical multiplexing plan comprising the DS-1 (1.544 Mb/sec), DS-1C (3.152 Mb/sec), DS-2 (6.312 Mbit/sec), DS-3 (44.736 Mb/sec) and DS4 274.176 Mb/sec signals.

Conventional circuit transmission systems suffer from a number of shortcomings. Perhaps the most important problem is the multiplexing hierarchy itself. An important result of the hierarchy is an inherent lack of flexibility. Since the network can only transmit the set of signals in the hierarchy, every telecommunications service has to meet the stringent interface requirement of given hierarchical signal bit rates, instead of the particular service being able to transmit at its own natural bit rate. Therefore, the packet mode of transmission which is inherently bit rate flexible is favored for future broadband networks which are to be adapted to deliver enhanced telecommunication services such as high definition video and interactive data communications.

In contrast with circuit transmission systems which transmit data in frames subdivided into slots, packet transmission systems transmit data in discrete blocks or packets, with each packet having an address header at the front thereof. At the network switches, packets are routed from a specific input line to a specific output line, based on address information contained in the packet header. In this way data packets can be routed from a particular subscriber location, through a telecommunications network, to another subscriber location. Packet transmission techniques and especially fast packet transmission techniques (see e.g., R. W. Muise et al., "Experiments in Wideband Packet Technology", Proc 1986 International Zurich Seminar on Digital Communications, pp. 136–138 are inherently bandwidth flexible (i.e. the number of packets generated by a given service per unit time is flexible) and thus are suitable for wideband enhanced communications services. Accordingly, it is desirable to introduce packet transmission technology

into the public telephone network, which up to now is based primarily on circuit transmission technology.

The commonly-held view as to how to introduce packet technology into the public network is to deploy a packet overlay network because the existing network is optimized for circuit transmission and is therefore incompatible with packet transmission techniques. Accordingly, many deployment strategies recommend overmeshing an overlay packet network for a set of wideband services and hope that the migration of new services to the packet overlay network will allow the existing circuit transmission network to be phased out slowly. The main advantage of a packet overlay network is the quick realization of an end-to-end network for new services. However, the approach requires a large initial capital investment and increases operational cost by requiring the management of multiple separate networks.

It is an object of the present invention to provide an alternate approach for introducing packet transmission technology into the public telephone network, which approach requires the replacement of existing transmission components but not the implementation of an entirely new network. Thus, it is an object of the invention to provide a digital data transmission system capable of handling both existing hierarchical circuit traffic and packet traffic.

With regard to the above-identified objects of the invention, it should be noted that recent advances in network switch designs have blurred the distinction between packet networks and circuit networks. A typical switch for use in a telecommunications network has three major components: control processor, switch interfaces, and interconnection network. The control processor handles call set-up and tear-down, maintenance and administrative functions. The switch interfaces convert transmission formats (i.e., the format data has when transmitted between switching nodes) to switch formats (i.e., the format data has when processed within switching nodes). The interconnection network routes information blocks from specific input lines to specific output lines of the switch. For the existing digital circuit systems used in the public telephone network, the information in a specific time slot on an incoming line is transferred, via the switch, to a specific time slot on an outgoing line. Thus, the interconnection network serves as a crossconnect for the incoming signals on a slot-by-slot basis.

It has recently been shown (see e.g., Day-Giacopelli-Huang-Wu, U.S. patent application Ser. No. 021,664 entitled Time Division Circuit Switch, filed on Mar. 4, 1987, now U.S. Pat. No. 4,782,474, issued Nov. 1, 1988, and assigned to the assignee hereof) that a switch for use in a circuit network can be built using a self-routing packet interconnection network. An example of such a self-routing packet network is the Batcher-banyan network. Based on the address headers associated with fixed sized packets, the Batcher-banyan network routes a plurality of packets in parallel to specific destination addresses (i.e., specific output lines) without internal collisions. Thus, to mimic the operation of the conventional time-space-time switches used in circuit networks, switch interfaces are provided which perform the time slot interchange function and which are able to insert headers in front of circuit slots to convert such slots into packets for routing through the self-routing interconnection network and able to remove headers from packets leaving the self-routing interconnection network to reconvert packets back into conventional circuit time-slot format.

In addition to circuit and packet transmission, another mode of digital transmission is known as Asynchronous Time Division Multiplexing (ATDM). See e.g., W. W. Chu "A Study of Asynchronous Time Division Multiplexing for Time Sharing Computer Systems" Proc AFIPS Vol 35, pp. 669-678, 1969 and A. Thomas et al. "Asynchronous Time Division Techniques: An Experimental Packet Network Integrating Video Communication" Proc International Switching Symposium, May 1984. ATDM is used in connection with continuous and bursty data traffic. ATDM uses channel identifiers with actual data to allow on-demand multiplexing of data from subscriber terminals with low channel utilization. The channel identifiers and associated data form time slots. However, ATDM is bit rate flexible since the appearance of packets can be asynchronous. Slot timing is obtained from a special synchronization pattern which is inserted into unused time slots. Since the synchronization pattern appears only in unused time slots, ATDM cannot be used to carry existing high speed hierarchical signals wherein the loading is close to one hundred percent.

In short, the situation is that the present public telephone network utilizes circuit transmission technology and the associated time division multiplexing transmission techniques, while future broadband services, the demand for which is presently uncertain, are best offered using packet transmission technology. It is therefore an object of the invention to provide a transmission system which is capable of integrating present circuit traffic with future packet traffic so as to provide a flexible migration strategy from the existing copper wire based circuit network to succeeding generations of high bandwidth packet transmission networks.

SUMMARY OF THE INVENTION

The digital network transport system of the present invention, referred to herein as Dynamic Time Division Multiplexing (DTDM), is a flexible network transport system capable of effectively handling both circuit and packet traffic. By combining conventional time division multiplexing techniques and packet transmission techniques, DTDM enables a flexible transition from the existing circuit type networks to future broadband packet transmission networks.

In a network utilizing DTDM, each transmission bit stream is divided into frames. These frames are the fundamental unit of data transport in DTDM. Each such frame comprises two fixed length fields: overhead and payload. The overhead field includes, for example, a frame alignment word for frame timing and the empty/full status of the frame. The payload field of each frame may be filled with a data packet including header or a slot from a circuit transmission stream. Before a slot from a circuit transmission stream can be inserted into the payload field of a DTDM frame, it must first be converted into a packet-like form with a header at its front. Viewed another way, each occupied DTDM frame comprises a transmission overhead field, a header field, and a data field. Thus, the DTDM transmission format is a combination of the circuit transmission format and the packet transmission format.

In the DTDM system, packet and circuit traffic can be multiplexed through the same multiplexer. Thus, such a multiplexer can have continuous circuit type tributaries and bursty packet tributaries. To multiplex

*669

such diverse traffic, a train of DTDM frames with empty payload fields is generated. This train has a bit rate which defines a basic backbone transmission rate for the DTDM transmission system. Data in the form of packets or circuit slots with headers attached are inserted into the empty frames to form the DTDM bit stream.

An appropriate analogy is as follows. The stream of empty DTDM frames may be analogized to a train of empty freight cars. The empty freight cars are then filled with data from the various tributaries which may have been in circuit or packet format.

Illustratively, a DTDM multiplexer may be used to merge traffic from three different communications sources or tributaries into a single DTDM bit stream. These tributaries may be a digital phone generating 64 Kilobits/sec PCM voice, a graphics terminal sending bursty data at 1 Megabit/sec, and a circuit transmission stream operating at the DS1 rate of about 45 Megabits/sec. Illustratively, the bit rate of the backbone DTDM bit stream is 150 Megabits which yields 144,000 frames per second given a 130-byte frame size. The available frames are shared by the three tributaries by giving higher priority to the circuit tributary, and allowing the voice and graphics tributaries to contend on a first-come, first-served basis. The circuit tributary seizes one out of every three empty frames passing by. Thus the regularity of the circuit transmission will be maintained throughout the DTDM transmission link. Illustratively, the voice source is packetized by accumulating up to 15 milliseconds worth of voice samples before inserting this information into an empty DTDM frame along with a header. In this case the voice tributary will on average seize one out of every 2,160 frames. Similarly, at a rate of 1 Megabit per second, the graphics tributary will fill one frame out of 150. In this way, three diverse data streams are multiplexed into a single bit stream.

As a second example, DTDM can be used as a replacement transmission technology to carry existing inter-office traffic. More specifically, consider the need to multiplex and transmit three hierarchical signals at the DS1, DS2, and DS3 rates, respectively, for point-to-point transmission between two offices. The traditional TDM approach would utilize a step-by-step hierarchical approach to multiplex and to subsequently demultiplex these signals. The conventional hierarchical multiplexing scheme requires line conditioning and synchronization circuitry at each level of the hierarchy as well as hardware for bit interleaving.

In contrast, using a DTDM multiplexer, time slots from each of the three signals would be inserted into the empty frames in a basic DTDM backbone signal. If the backbone signal is 150 megabits per second and comprises 144,000 frames per second, the DS3 signal would require one out of every three DTDM frames, the DS2 signal would require approximately one out of every twenty-one DTDM frames and the DS1 signal would require approximately one out of every eighty-four of the empty DTDM frames.

In an actual network, the above-described DTDM streams at the basic backbone bit rate generally contain empty frames; thus DTDM streams may be multiplexed into more densely populated DTDM bit streams at the same bit rate. These more densely populated basic backbone rate bit streams may then be multiplexed into higher bit rate streams for point-to-point inter-office transmission.

Details of the assemblers needed to form the basic DTDM bit streams, the disassemblers needed to disassemble the basic DTDM bit streams, and the set of multiplexers and demultiplexers needed to implement DTDM in an actual network are described in detail below along with a framer circuit which plays a significant role in particular implementations of the assemblers/disassemblers and multiplexers/demultiplexers.

BRIEF DESCRIPTION OF THE DRAWING

FIG. 1 schematically illustrates the DTDM transmission format, in accordance with an illustrative embodiment of the invention;

FIG. 2 schematically illustrates the formation of a backbone DTDM, bit stream, in accordance with an illustrative embodiment of the invention;

FIG. 3 schematically illustrates an end-to-end network using DTDM, in accordance with an illustrative embodiment of the invention;

FIG. 4 illustrates an assembler for combining diverse tributary data streams into a single DTDM stream, in accordance with an illustrative embodiment of the invention;

FIG. 5 illustrates a disassembler for separating a DTDM bit stream into diverse tributary data streams, in accordance with an illustrative embodiment of the invention;

FIG. 6 illustrates a multiplexer for combining a plurality of DTDM bit streams into a single more densely occupied DTDM bit stream having the same bit rate;

FIG. 7 illustrates an N:M multiplexer for combining a plurality of DTDM bit streams;

FIG. 8 illustrates how the input lines in the multiplexer of FIG. 7 are grouped;

FIG. 9 illustrates a demultiplexer for separating a densely occupied DTDM bit stream into a plurality of less densely occupied DTDM bit streams;

FIG. 10 illustrates a multiplexer for point-to-point transmission.

FIG. 11 illustrates a demultiplexer for use in connection with point-to-point transmission; and

FIG. 12 illustrates a framer circuit.

DETAILED DESCRIPTION

1. DTDM Transmission Format

DTDM is an approach to data transport which can handle both TDM hierarchical signals and packet traffic in a common integrated structure, while allowing complete bit rate flexibility. As illustrated in FIG. 1, the transmission bit stream is divided into frames 1. The DTDM frame is the fundamental unit of information transport in the DTDM transmission scheme. The frames come one after the other so as to form a continuous chain or train.

Each frame 1 comprises two fixed length fields designated transmission overhead (T) and payload in FIG. 1. Illustratively, each frame comprises 130 bytes with 10 bytes being allocated to the transmission overhead field. Typically, the bit rate of the DTDM bit stream illustrated in FIG. 1 is about 150 Megabits/sec. The following information may be available in the overhead field of every DTDM frame; frame alignment word for frame timing, empty/full status of the frame, and span identification.

As shown in FIG. 1, the payload field of each frame may be filled with a data packet including a header (H) or a slot from a circuit transmission stream. However,

*670

before a slot from a circuit transmission stream can be inserted into the payload field of a DTDM frame, it must first be converted to packet-like form by the insertion of a header (H) at its front. Viewed another way, each occupied DTDM frame comprises a transmission overhead field, a header field, and an information field. Thus, the DTDM transmission format is a combination of the circuit transmission format and the packet transmission format. The packet header provides information such as channel number, line number, error detection, etc. In general, only the information required in every frame gets permanent bandwidth allocation in the transmission overhead field.

FIG. 2 schematically illustrates the formation of a DTDM bit stream. The DTDM bit stream assembler 3 can combine into a single bit stream both continuous circuit tributaries and bursty packet tributaries. Three such tributaries are illustrated in FIG. 2. They are: a digital phone tributary 5 generating 64 Kilobits/sec PCM voice, a tributary 7 from a graphics terminal sending bursty data at one megabit per second, and a circuit transmission stream 9 operating at the DS3 rate of about 45 Megabits/sec. Each of the three tributaries has a characteristic shading in FIG. 2 so that it is possible to follow how data from the three tributaries is combined to form the DTDM bit stream.

To multiplex such diverse traffic, a train 10 of DTDM frames with empty payload fields is generated. This train 10 has a bit rate which defines a basic backbone transmission rate for the DTDM system. Each of the frames in the train 10 has an occupied transmission overhead field (T).

Illustratively, the train of frames has a bit rate of about 150 Megabits per second and comprises 144K blocks/sec. The assembler 10 serves to insert data from the tributaries 5,7,9 into the payload fields of the DTDM frames in the stream 10. To accomplish this, the tributaries 5, 7, 9 are first packetized using packetizers 11, 13, 15, respectively to form the packetized streams 17, 19, 21. Each packet comprises a header (H) and an information field. In the case of the tributary 5, up to 15 milliseconds of speech samples are accumulated to form a packet. In the case of the circuit tributary each slot is converted to packet form by placing a header at the front thereof.

To form the DTDM stream 10, the packets comprising the streams 17, 19, 21 are inserted into the empty payload fields of the empty frames in the stream 10. The empty frames are shared by the three tributaries by giving higher priority to the circuit tributary 9 and allowing the voice and graphics tributaries 5, 7, to contend for empty frames on a first-come, first-served basis. Thus, the circuit tributary seizes one out of every three frames so that the regularity of the circuit transmission is maintained throughout the DTDM transmission link. Similarly, the voice tributary will seize one out of every 2,160 frames and the graphics tributary will seize on average one out of every 150 frames. It should be noted that the bit stream 12 is not 100% occupied and that some frames remain empty. In this way, three diverse tributaries are multiplexed into a single DTDM bit stream.

2. A Network Utilizing DTDM

FIG.3 schematically illustrates an end-to-end network 20 utilizing DTDM. The network 20 connects to customer premises equipment (CPE) 22, of which three types are illustrated, namely video, voice and data.

In the network 20, three multiplexing stages are required to support end-to-end transport. In the user-network interface stage 30, an assembler 32 receives data streams on lines 21 from the customer premises equipment 22 and combines these streams into a basic backbone DTDM stream of the type discussed in connection with FIGS. 1 and 2. Similarly, disassembler 34 tears apart a basic DTDM bit stream arriving on line 33 and distributes the data to the appropriate customer premises equipment 22 via lines 23.

As indicated above, the DTDM bit stream formed by the assembler 32 is not 100% occupied. Thus the multiplexer 36 in the remote electronics stage 38 is used to combine several DTDM bit streams arriving on lines 62 into a more densely occupied DTDM bit stream of the same bit rate to achieve greater transmission efficiency. Similarly, the demultiplexer 39 separates a densely populated DTDM bit stream arriving on line 120 into less densely populated DTDM bit streams transmitted via lines 33, so that the data contained therein can ultimately be routed to the correct customer premises equipment.

In the point-to-point stage 40, a plurality of DTDM bit streams arriving via lines 63, 150 are time division multiplexed by means of time division multiplexer 42 for high speed point-to-point transmission via line 165 to a network switch (not shown). For example, the multiplexer 42 receives one DTDM stream via a line 63 from multiplexer 36 and another DTDM stream via line 150. The DTDM bit stream transmitted via line 150 is formed by DTDM assembler 43 and contains the data of three DS3 tributaries 45.

Time division demultiplexer 44 receives a high speed bit stream from a switch (not shown) via line 170 and demultiplexes this stream into a plurality of DTDM streams. One DTDM stream containing data for customer premises equipment goes to demultiplexer 39 via line 120 and another DTDM stream comprising DS3 slots goes to disassembler 47 via line 179.

3. DTDM Assembler and Disassembler

The function of the DTDM bit stream assembler 32 of FIG. 3 is to packetize each incoming data stream associated with one particular customer service or transmission channel and then embed these packets into the basic DTDM transmission frames. The assembler 32 is shown in greater detail in FIG. 4.

The assembler 32 comprises a plurality of interface units 50. Each interface unit 50 serves to interface an associated data input 21 with the DTDM bit stream. A DTDM bit stream comprising empty frames with empty payload fields is generated by framer unit 52. A detailed description of the framer unit is provided below.

Each interface unit includes a framer unit 53. The framer units 52, 53 are connected together in a daisy chain fashion. The frames comprising the DTDM bit stream are passed along the daisy chain from one framer unit to the next. More particularly, the DTDM bit stream leaves the serial data output (sdo) of the framer unit 52 and enters the serial data input (sdi) of the topmost framer unit 53. The DTDM bit stream leaves the topmost framer 53 via its serial data output (sdo). The DTDM bit stream then enters the serial data input (sdi) of each succeeding framer unit and leaves via the serial data output (sdo) of each framer unit. The DTDM bit stream leaves the serial data output of the lowermost framer via line 62. As shown in FIG. 3, line 62 serves to

transmit the DTDM bit stream to the DTDM multiplexer 36. If the DTDM frame currently located at the framer unit 53 of a particular interface 50 is empty, that interface may insert a packet into the payload field of the DTDM frame.

The data inputs 21 to the assembler 32 are connected to the customer premises equipment 22 of FIG. 3 and may have a wide range of bit rates; for example, the data inputs 21 can be video, voice, data, or different digital hierarchical transmission signals (DS-1, DS-2 and DS-3). Therefore, the assembler architecture must be capable of efficiently accommodating different input bit rates and be flexible enough to allow for future expansion or for the changing of particular input connections to different services. The architecture shown in FIG. 3 provides the capability to easily add or drop a particular input service.

Each input 21 is connected to a packetizer 55 which forms part of the associated interface unit 50. The packetizer 55 puts the incoming data into a packet structure by adding a packet header at the beginning of appropriate segments of the input bit stream. The packet header carries information about the packet, such as packet occupancy, channel identification number, line identification number, check sum and so on. Illustratively, the channel identification number is used to identify the input service from which the packet originated. After the data is put into a packet structure, it is stored in a FIFO 57 with byte wide format. The framer unit 53 then reads the data from the FIFO 57 into its parallel data input (pdi) 58 and generates properly framed data bits which are inserted into an empty payload field of a DTDM frame currently at the particular framer unit 53.

However, a framer unit 53 will not read the data from the FIFO 57 unless two conditions are met. One is that the "packet-ready" pulse signal from the packetizer 55 is asserted, indicating one packet is completely stored in the FIFO. The other condition is that the incoming DTDM frame on the serial data input (sdi) of the framer 53 is not already occupied by a valid packet, i.e. the incoming DTDM frame is empty. Thus, an empty or "emp" signal is transmitted from the framer 53. The "packet-ready" signal triggers an enable signal, "en", in the framer unit to be asserted for the whole frame transmission period allowing the data packet to be moved from the FIFO 57 through the framer 53 and into the DTDM bit stream. Using the "emp" and "en" signals, control logic 59 controls the reading of a packet out of the FIFO 57 and into the framer 53.

Since the framer units 53 are daisy-chained together, the contention for empty DTDM frames is automatically resolved in favor of input services having positions closer to the empty frame generator.

In order to simplify the assembler 32 of FIGS. 3 and 4 and hence reduce the building cost, one practical assumption may be utilized; the total traffic of all inputs at any given time is less than the bit rate of the basic backbone DTDM stream.

The topmost framer 52 in FIG. 4 does not have any input service connected to it. It generates the chain of empty DTDM frames which are sent to the following framers 53. If none of the interfaces 50 insert a packet into a particular frame, an empty frame is finally sent out through the serial data output (sdo) of the bottommost framer unit on lead 62.

After the DTDM bit stream has traveled through the entire communications network 20 of FIG. 3, which network includes multiplexers, switches, and demultiplexers, etc., the DTDM bit stream is disassembled and the data distributed to the appropriate customer services equipment. In the network 20 of FIG. 3, the disassembler 34 is used for this purpose. The disassembler 34 is shown in greater detail in FIG. 5. Illustratively, the disassembler 34 removes both the transmission overhead and packet header field from each incoming DTDM frame and distributes the data contained in the frame to the desired customer premises device.

More particularly, the assembler 34 comprises a plurality of interfaces 66. Each interface 66 receives the incoming DTDM bit stream via line 33 (see FIG. 3) and is illustratively connected to one customer premises device via an output lines 23 (see FIG. 3). Each incoming DTDM frame is simultaneously received by the framer unit 70 in each interface 66. However, only packets containing data to be transmitted to the associated customer premises equipment are transferred from the framer 70 to the associated FIFO 72. To accomplish this, the packet occupancy and channel identification number are examined by the framer 70. The framer 70 in turn generates proper control signals via lines 73, which, along with control logic 74, determine whether or not the packet carried in the payload field of the particular DTDM frame will be written into the FIFO 72 of the particular interface unit so that the data contained in the packet can be transmitted to the associated customer premises equipment.

Recovering the correct frequency from the incoming data is a very challenging task. Although for each kind of customer premises equipment or service the frequency is known, the difference between the local reading clock used to read data out of the FIFO 72 and the clock which was used to load data into empty frames at the transmit end may result in overflow or underflow of the FIFO 72. Illustratively, a phase locked loop 78 is used to modify the local reading clock in order to cancel this difference in clock rates.

As shown in FIG. 5, the local reading clock signal (line 73r) used to read data out of the FIFO 72 is phase locked with the incoming data so that the data can be read out correctly from the FIFO 72 without overreading or underreading. The rate at which data is read out of the FIFO is determined by the frequency of the voltage controlled oscillator 76 in the phase locked loop 78.

The packet is written into the FIFO 72 with the network clock rate, but read out at a rate dependent on the particular equipment to which the data is transmitted. An "hf" signal which indicates that the FIFO 72 is half full is smoothed out by a low-pass filter 75 whose output is used to control the output frequency of a voltage-controlled oscillator 76. If information is read out of the FIFO 72 faster than information is written into the FIFO 72, then the "hf" signal will not be asserted. This causes the voltage output from the low-pass filter 75 to decrease, reducing the output frequency produced by the voltage controlled oscillator and reducing the rate at which data is read out of the FIFO 72. Similarly, if information is read out of the FIFO more slowly than it is being written into the FIFO 72, the "hf" signal will be asserted and the voltage controlled oscillator frequency will be increased so that the read clock signal frequency is larger. The same interface unit 66 can be used for different customer premises devices by choosing a proper frequency for oscillator 69.

Data packets read out of FIFO units 72 are depacketized by means of depacketizer circuits 79 which serve to remove the headers. The resulting data is then trans-

*672

ruled via lines 23 to the appropriate customer premises equipment.

4. DTDM Bit stream Multiplexer and Demultiplexer

The function of the DTDM bit stream multiplexer 36 of FIG. 3 is to concentrate a plurality of relatively sparsely occupied incoming DTDM streams into at least one more densely occupied DTDM stream of the same bit rate, resulting in more efficient use of the transmission facility. There is more than one architecture for implementing the DTDM multiplexer 36 of FIG. 3.

One embodiment of such a DTDM bit stream multiplexer is illustrated in FIG. 6. The DTDM multiplexer 36 of FIG. 6 comprises N input lines 62 (see FIG. 3) and one output line 63 (see FIG. 3). Line buffers 64 recognize and queue incoming DTDM frames. The server 65 looks for newly arrived DTDM frames in the line buffers 64, adds a proper line number in the header field, and sends the frames out in a more densely occupied DTDM bit stream.

The primary functions of the line buffers 64 are recognition and queuing of incoming DTDM frames. Each line buffer contains a serial/parallel converter 66 for converting incoming serial DTDM frames into parallel form and a first-in, first-out buffer 67 with capacity for multiple frames. A timing and control circuit 68 operates the line buffer and interfaces it with the server.

The main functions of the server 65 are to look for newly arrived DTDM frames in the line buffers, to modify the header field to include a line number, and to place the DTDM frame in a more densely occupied DTDM bit stream. The server comprises a header insert circuit 69 for modifying the header field of the DTDM frames, a controller circuit 70 for interfacing with the line buffers 64, and a parallel to serial converter 70. The operations of the server are pipelined; while the server reads a DTDM frame from a line buffer and places it in an outgoing DTDM stream, it continues searching line buffers for DTDM frames. It should be noted that the multiplexer of FIG. 6 is useful for multiplexing packets in non-DTDM transmission formats in addition to being useful for DTDM bit streams.

Another possible architecture for a multiplexer capable of combining several relatively sparsely occupied DTDM bit streams into a more densely occupied DTDM bit stream of the same bit rate builds on the architecture of the DTDM bit stream assembler 32 of FIG. 4. Each input to an interface unit 50 of FIG. 4 is replaced by a serial data link on which a DTDM bit stream arrives. The data packets contained in the frames comprising the incoming DTDM bit stream contend for output frames in an outgoing DTDM bit stream. The frames comprising the outgoing DTDM bit streams are generated by the framer 52 and passed along the chain of interconnected framers 53. The interface units 50 insert data packets from incoming DTDM frames into the frames of the outgoing bit stream to form a more densely occupied DTDM bit stream. The contention for output frames is resolved automatically by the daisy-chained connection of the framer units. Note that no packetizer is needed in the interface units, and the length of each FIFO is preferably more than two frames to prevent data packets contained in incoming frames from being lost.

FIG. 7 schematically illustrates an alternative DTDM bit stream multiplexer for combining a plurality of relatively sparsely occupied DTDM bit streams into a smaller number of more densely populated DTDM bit streams of the same bit rate. The multiplexer 36' of FIG. 7 has the flexibility to receive N input DTDM bit streams and to transmit M output DTDM bit streams, which allows M output lines to be shared by N input lines. It is known that both the probability of buffer overflow and the average delay for bursty traffic can be significantly decreased by increasing the number of outputs.

Using the multiplexer architecture 36 shown in FIG. 6, it is difficult to build an N:M multiplexer, because the service order is determined by a single central server. However, it is possible to provide a multiplexer system, comprising M separate multiplexers of the type shown in FIG. 6, each having N/M input lines and line buffers and one server and associated output line. In contrast, the service order in the multiplexer 36' of FIG. 7 is determined locally, which results in the flexibility of reassigning input lines to different output lines based on the input traffic statistics.

The N:M multiplexer 36' of FIG. 7 comprises a plurality of input lines 62 (see FIG. 3) and a smaller number of output lines 63 (see FIG. 3). DTDM frames arriving on the input lines 62 are converted into a byte wide stream by means of the serial-to-parallel converters 86 and stored in the associated buffers (FIFOs) 88. The operation of the framer units 90, 92 is similar to those in the DTDM bit stream assembler of FIG. 3. Each framer 90, 92 has a parallel data input (pdi), a serial data input (sdi) and a serial data output (sdo). Framer 90, the head-end framer unit, doesn't have any input lines connected to it. In normal operation, it continuously sends out a chain of empty frames. The remaining framers 92 take data comprising occupied DTDM frames in the buffers 88, and insert this data into the empty frames generated by the framer 90, so as to combine a plurality of sparsely occupied DTDM bit streams into a smaller number of more densely populated DTDM bit streams.

The N:M multiplexer 36' of FIG. 7 comprises an $(N+M)\times(N+M)$ broadcasting cross point switch network 95. The serial data output (sdo) of each framer 90, 92 is connected to an input of the switch, and the serial data input (sdi) of each framer 9 is connected to an output of the switch as shown in FIG. 7. The connections through the switch network are controlled by a dedicated controller 97.

Illustratively, when the system is initialized, the N input lines 62 are divided into M groups, $(1,2 \ldots, J)$, $(J+1, J+2, \ldots, 2J), \ldots, (MJ+1, \ldots, (M+1)J)$, where $J=N \bmod M$. The J input lines in each group are logically connected as shown in FIG. 8. Each group of input lines is associated with one output line. Thus, all of the DTDM frames arriving at the inputs of one group are merged into a single DTDM bit stream which leaves via the associated output. The topmost framer unit 92 in each group receives empty frames broadcast from the framer 90. Each frame is then passed through the switch 95 from one framer in the group to the next framer in the group. If a particular FIFO 88 has data comprising a DTDM frame and the associated framer 92 receives an empty frame, the data is inserted into the empty frame. Thus, within each group service priority is ranked in descending order with the higher priorities near the top. Ultimately, M relatively densely occupied DTDM bit streams leave the multiplexer of FIG. 6 via the outputs 63.

Thus, with the addition of the cross point switch, more than one framer 92 receives empty frames from the framer 90 at the same time. This achieves the N:M

*673

multiplexing function automatically and with minimal complexity.

If the input lines are not grouped so as to distribute output traffic evenly, the input lines can be regrouped easily by changing the connections within the switching network 95. For example, a particular input in the first group of inputs may be assigned to any other group, e.g., the second group of inputs, to spread out traffic evenly. The controller 97 must know the traffic statistics of each input line and follow some algorithm to rearrange the inputs and decide the ordering (priority) within each group.

The DTDM multiplexer of FIG. 7 may route DTDM frames arriving on the same input line to different output lines. For example, n frames arriving on an input line have been sent to output #1. But the (n+1)th frame may be switched to output #2 because reconfiguration took place to balance traffic among the output lines. This may cause an out of sequence problem if the (n+1)th frame arrives at the receive end before the nth frame does. The cost to reorder the frame sequence at the output end may be high. Illustratively, to avoid this problem, one rule may be followed: the input lines carrying services with high bit rate information, such as video, will not be switched from one input line group to another during the service period. For a low bit rate service, such as voice at 64 Kb/s, even if two consecutive frames containing data are dispatched onto two different output lines, the two frames from such a bursty service will be separated by more than several hundred frame intervals. Hence, it is unlikely for there to be an out of sequence problem in this case.

It should be noted that multiplexer architecture of FIG. 7 may be used to multiplex other types of traffic besides DTDM traffic. For example, streams of data packets may be multiplexed together to form more densely occupied streams.

Turning now to FIG. 9, the DTDM bit stream demultiplexer 39 of FIG. 3 is illustrated in greater detail. The function of the DTDM bit stream demultiplexer 39 is to separate a relatively densely occupied incoming DTDM bit stream into a plurality of relatively sparsely occupied outgoing DTDM bit streams of the same bit rate so that the user data in the frames can ultimately be transported to the proper customer premises devices.

The demultiplexer 39 of FIG. 9 has one input line 120 (see FIG. 3) and a plurality of output lines 33 (see FIG.3). Each output line 33 is connected to the input line 120 by means of an associated interface 124. Any incoming DTDM frame is simultaneously received by the framer unit 126 in each interface 124. The frame occupancy and line identification number of each incoming DTDM frame are examined by the framers 126. If the frame is not empty and the line number is matched, the packet contained therein will be written into the FIFO 130 under the control of logic 128 and then read out of the FIFO 130 by the framer 132 at the output end of the interface 124 under the control of logic 134. Otherwise, the packet is simply discarded. In this manner, the data from each incoming frame is routed to the correct output line. A counter 136 in each interface is used to count the number of bytes written into the FIFO 130 and generates a signal when a full packet is stored in the FIFO. This signal will inform the output framer 132 to start reading the packet in the FIFO. The framer 132 will assert the "en" signal during the reading of the entire packet. The framers 132 generate sequences of DTDM frames. These sequences of frames leave the framer 132 via the serial data outputs and form the outgoing DTDM bit streams on the lines 33. When data packets are present in the FIFOs 130 they are inserted into the frames generated by the framers 132. In particular embodiments of the demultiplexer, the functions of the framer units 126, 132 may be performed by a single framer unit.

5. Time Division MUX/DEMUX for DTDM Bit Stream

After relatively sparse DTDM bit streams are concentrated into more densely populated DTDM bit streams of the same bit rate using for example, the DTDM multiplexer 36 of FIG. 3, a plurality of such more densely populated bit streams may be time division multiplexed into a higher speed data stream using, for example, the time division multiplexer 42 of FIG. 3. Such high speed data channels may be used for communications to and from central offices.

Usually, the most challenging work in a time division multiplexing system is to synchronize all incoming bit streams so that they have a common bit rate before they are interleaved into a higher bit rate stream. Typically, the input bit streams have the same nominal center frequency but drift independently a small amount from the center frequency. The conventional way to overcome the asynchronization among the input bit streams is positive bit or byte stuffing. The frequency of the high speed output bit stream is made greater than the product of the nominal center frequency and the number of input tributaries. There is usually a bit or byte position reserved for the occasional stuffing of a dummy bit or byte. Also, there is some control overhead used to indicate if the bit or byte at the stuffing position is valid.

By taking advantage of the fact that the frames comprising each input DTDM bit streams are not 100% occupied, the frequency of the higher speed output bit stream can be made exactly equal to the nominal center frequency of the input tributaries times the number of the input tributaries. In the case of a DTDM system, this can be accomplished through the positive and negative stuffing of DTDM frames. Since the frequency of each input tributary signal can be adjusted in the positive or negative direction through the insertion or removal of an empty DTDM frame, it is possible to make the frequency of the high-speed multiplexed bit stream exactly an integer multiple of the nominal input tributary frequency.

A time division multiplexer 42 (see FIG. 3) for multiplexing a plurality of DTDM bit streams is illustrated in FIG. 10. Each input 63, 150 (see FIG. 3) is connected to an interface unit 152. Each interface unit 152 comprises a framer 154 which is clocked by a clock signal derived from a clock recovery circuit 156. The derived clock, which is the actual frequency of the tributary, may differ slightly from the nominal tributary frequency as discussed above. This difference between the nominal and actual frequencies is eliminated in the interface unit. Each incoming DTDM frame will be examined by the framer 154 in the associated interface 152 for its occupancy. The data packets contained in the occupied frames will be written into the FIFO 158 under the control of logic 159 and read out later by the framer 160 at the output end of the interface unit 152. Empty frames are discarded.

The reading of the data packets from the FIFOs 158 to the parallel data inputs (pdi's) of the framers 160 is synchronized. The serial data input (sdi) of each framer

160 is connected to the serial data output of a framer 162. The framer 162 serves to broadcast empty frames to the framers 160 so that each framer 160 receives a synchronous chain of empty frames at the nominal tributary frequency. The empty frames received by each framer unit 160 are filled with data packets from the associated FIFO 158 to produce synchronized tributary bit streams at the nominal tributary frequency.

If the actual frequency of a particular tributary is less than the nominal center frequency then on occasion, the associated FIFO 158 will not have a packet to insert into an empty DTDM FRAME. The net effect is that an empty DTDM frame is added so that the tributary acquires a frequency equal to the nominal frequency. However, if the actual frequency of the tributary is larger than the nominal center frequency the net effect is that empty DTDM frames are dropped so that the tributary acquires a frequency equal to the nominal frequency. Illustratively, the difference between the actual and nominal tributary frequencies is on the order of ten parts per million. In this case, a two frame capacity FIFO 158 is sufficient as long as each input tributary has one empty frame in 10^3.

All of the framers 160 send out frames at the same time, with frame alignment being automatically achieved. The aligned frames are then bit interleaved using bit interleaving circuit 164 to produce a single high bit rate bit stream at output 165 (see FIG. 3). Note that the clocks of the framers 160 are connected together so that data bits coming from the framers 160 are phase aligned and can be bit interleaved directly. The clock for the framers 160 is provided by the clock generator 166 and frequency divider 167. In an alternative embodiment of a time division multiplexer, instead of bit interleaving, frame or byte interleaving may also be used. If the frame interleaving is used then the multiplexed output bit stream has the same DTDM frame structure, thereby allowing the flexible single transport architecture to grow as the technology advances.

A time division demultiplexer 44 (see FIG. 3) for demultiplexing the high speed bit stream is illustrated in FIG. 11. The high speed data stream arrives on input line 170 (see FIG. 3) and is bit deinterleaved by means of bit deinterleave circuitry 172 into several lower speed tributary bit streams which are transmitted outward on lines 174. In order to dispatch the bits to correct tributaries, a predetermined span identification (SPID) is inserted for each tributary before they are multiplexed at the transmit side. The tributary present on line 174$_3$ is connected to a framer unit 176, which will detect the frame boundary and determine by examining the span identification whether or not the bit deinterleave circuitry has correctly aligned the incoming bit stream so that appropriate data goes to appropriate output tributaries. If not, either a skip pulse is generated to rotate the bit sequence or a signal is generated by the framer 176 and sent to a cross point switch 178 to reassign the order of the bit stream. The bit streams with correct bit assignments appear at outputs 120, 179 (see FIG. 3). Alternatively, instead of the crosspoint switch 178, a barrel shifter may be used. It should be noted that the clock for the bit deinterleave circuit 171 and framer 176 is provided by clock recovery circuit 180 and frequency divider 182. Demultiplexers which operate according to similar principles are disclosed in R. J. Boehm et al. "Standardized Fiber Optic Transmission Systems - A Synchronous Optical Network View" IEEE Journal on Selected Areas in Communications VOL SAC-4 No. 9 pp 1474–1431 Dec. 1986 and L. R. Linnell "A Wide-band Local Access System using Emerging-technology Components" IEEE Journal on Selected Areas in Communications" VOL SAC-4 No. 4 pp 612–618 July 1986.

6. The Framer Circuit

The framer unit is an important component for the implementation of specific embodiments of the assemblers, disassemblers, multiplexers and demultiplexers which comprise the DTDM network discussed above.

The framer unit performs a number of functions in the DTDM network, including generating trains of empty DTDM frames, enabling the writing of data packets into specific DTDM frames, and the examination of header data in specific DTDM frames to generate signals for the control of peripheral circuits (e.g. in a DTDM demultiplexer to determine if data in a particular DTDM frame belongs to particular customer premises equipment of a particular less densely occupied DTDM bit stream). All of these functions may be carried out by the framer unit discussed below.

A framer unit 200 is schematically illustrated in FIG. 12. Illustratively, the framer unit 200 is formed as a single chip. The framer unit 200 has a serial data input 202, a parallel data input 204, a serial data output 206 and a parallel data output 208. Timing information for the framer unit 200 is provided by timing generator 209. The framer 200 operates under control of a control unit 210 which illustratively comprises one or more finite state machines.

As indicated above, a plurality of framer units may be connected in a daisy chain fashion and DTDM frames may be passed from one framer to the next (see e.g., framers 53 of FIG. 4). Data may be written into an empty DTDM frame as follows. A DTDM frame is received at the serial input 202. The DTDM frame is converted to parallel form by serial-to-parallel converter 212 and is detected by frame detector 214. The frame detector 214 is in communication with the control 210 and illustratively communicates to the control 210 information such as whether or not the frame is empty. Illustratively, the DTDM frame leaves the framer unit via the serial output 206 after conversion to serial form by way of parallel-to-serial converter 216. However the frame cannot reach the parallel-to-serial converter 216 unless the control 210 applies a signal to the tristate device 218.

The data to be written into the frame is received at the parallel data input 204 (illustratively from a FIFO 57 in the DTDM bit stream assembler 32 of FIG. 4). If the particular DTDM frame is empty and data is available at the parallel input 204, a signal is applied by the control 210 to the tristate device 220 to enable the data to be inserted into the particular DTDM frame via bus 219 before it leaves the framer unit. However, if the DTDM frame is already full the control does not provide such a signal to the tristate 220. In particular framer units additional information such as span identification may be inserted into specific DTDM frames by means of an additional tristate unit not shown.

The framer unit 200 may also be utilized to generate a chain of empty DTDM packets (see e.g., framer 52 in FIG. 4). In this case the serial input 202 and associated serial-to-parallel converter 212 are not utilized. Instead, the control 210 applies a periodic signal to tristate 222 so that a frame alignment word is periodically read from frame byte ROM 224 and transmitted via bus 219 to

*675

parallel-to-serial converter 216 and serial output 206 so as to define a train of empty DTDM frames. Other information comprising the transmission overhead (T) field of the DTDM frame may also be stored in ROM 214 or provided by other sources connected to the bus 219 via a tristate device operative under the control of the control unit 219.

In particular situations (see e.g., framers 70 of FIG. 5 and 126 of FIG. 9), a framer unit receives occupied DTDM frames and the header (H) or transmission overhead (T) fields have to be examined to control peripheral circuit operations such as the reading of data into a FIFO. In this case, a multiple byte delay unit 230 may be included in the path between the serial input 202 and the parallel and serial outputs 208, 206. Typically a frame arrives at the serial input 202 and is converted to parallel form by the serial-to-parallel converter 212. The frame detector detects the frame and supplies necessary information from the header or transmission overhead fields to the control unit 210 which issues appropriate control signals via lines 232 such as user read/write strobes. Illustratively, the user read/write strobes control the writing of data from DTDM frames in the framer unit into associated FIFOs or other buffers. If the FIFO has byte wide format, the parallel output 206 may be used for this purpose. The delay unit 230 is used to insure that the necessary signal processing takes place before the DTDM frame leaves the framer unit.

7. Conclusion

A data transmission technique known as Dynamic Time Division Multiplexing (DTDM) has been disclosed along with an end-to-end network utilizing DTDM.

Finally, the above described embodiments of the invention are intended to be illustrative only. Numerous alternative embodiments may be devised by those skilled in the art without departing from the spirit and scope of the following claims.

What is claimed is:

1. A method for simultaneously transmitting data from sources having different bit rates in a telecommunication network comprising the steps of:

generating a bit stream comprising a sequence of frames, each of said frames including a transmission overhead field containing frame timing information and an empty payload field, and

filling the empty payload fields in said frames with data in packetized format from a plurality of sources which have access to the bit stream including circuit or packet sources, such that data in packetized format from any of said sources is written into any available empty payload field of any of said frames for transmitting data from each of said sources at its own desired bit rate via said bit stream and for transmitting data from said plurality of sources simultaneously via said bit stream.

2. The method of claim 1 wherein prior to filling said frames with slots from a circuit transmission stream, said slots are converted to said packetized format by placing a header in front of each of said slots.

3. A method for generating a bit stream capable of transporting data originating from both circuit transmission and packet sources comprising

generating a bit stream comprising a sequence of frames, each of said frames including a transmission overhead field containing frame timing information and an empty payload field,

packetizing data from a plurality of sources having different bit rates and which have access to said bit stream including circuit transmission sources or customer premises equipment to produce data packets, and

inserting said packets from said sources into the empty payload fields of said frames such that a packet from any of said sources is inserted into any available empty payload field of any of said frames for transmitting data from each of said sources at its own desired bit rate via said bit stream and for transmitting data from said plurality of sources simultaneously using said bit stream.

4. An apparatus for assembling a dynamic time division multiplexing bit stream comprising,

generating means for generating a train of frames wherein each frame includes a transmission overhead field containing timing information and an empty payload field,

processing means for processing data from a plurality of sources into packet format, and

inserting means for receiving said train of frames and for inserting each of said packets comprised of data from one of said plurality of sources into any empty payload field of any of said frames available to said inserting means to form said bit stream so that data from each of said sources can be transmitted at its own desired bit rate via said bit stream and so that data from said plurality of sources can be transmitted simultaneously via said bit stream.

5. The apparatus of claim 4 wherein said sources include circuit transmission bit streams or customer premises equipment.

6. An apparatus for assembling a bit stream for transmitting data from a plurality of sources comprising:

means for generating a train of frames, each of said frames including a transmission overhead field and an empty payload field, and

a plurality of interfaces, each of said interfaces serving to interface one of said sources with said train of frames, each of said interfaces comprising:

packetizing means for converting data into data packets,

memory means for storing at least one of said packets formed by said packetizing means, and

circuit means for inserting a packet stored in said memory means into any empty payload field of any available one of said frames so that data from each one of said sources can be transmitted at its own desired bit rate via said bit stream and so that data from said plurality of sources can be transmitted simultaneously via said bit stream.

7. The apparatus of claim 6 wherein said interface units are connected to one another serially and wherein said frames are passed sequentially to each of said interface units to receive said packets in said empty payload fields.

* * * * *

*676

United States Patent [19]
Hubbard et al.

[11] Patent Number: 4,835,768
[45] Date of Patent: May 30, 1989

[54] HIGH SPEED DIGITAL SIGNAL FRAMER-DEMULTIPLEXER

[75] Inventors: William M. Hubbard, Middletown; Dennis T. Kong, Holmdel, both of N.J.

[73] Assignee: Bell Communications Research, Inc., Livingston, N.J.

[21] Appl. No.: 181,560

[22] Filed: Apr. 14, 1988

[51] Int. Cl.[4] H04J 3/06
[52] U.S. Cl. 370/106; 375/116
[58] Field of Search 370/106, 105, 100; 375/114, 116

[56] References Cited

U.S. PATENT DOCUMENTS

3,597,547	8/1971	Roll	370/106
3,798,378	3/1974	Epstein	370/106
4,371,908	6/1983	Schenk	370/106
4,665,533	5/1987	Tomikawa	370/106
4,719,624	1/1988	Bellisio	370/100

Primary Examiner—Douglas W. Olms
Attorney, Agent, or Firm—James W. Falk; Lionel N. White

[57] ABSTRACT

A framer-demultiplexer circuit provides means for reducing the high serial bit-stream rate of byte-interleaved low level signal frame structures proposed by the Synchronous Optical Network (SONET) signal hierarchy to speeds which can be processed with low-power low-cost CMOS VLSI technology, while establishing and maintaining basic byte integrity. In this circuitry the incoming high-rate serial bit stream is divided alternately between shift registers 43, 44 under the control of a single high-precision clock-division circuit to provide a multi-bit formatting that enables parallel delivery of stage bytes with the multifold reduction in transmission to a rate within the processing capabilities of CMOs devices. Necessary synchronization of the register and latching elements of the circuit with the incoming bit stream is effected through use of comparator means 62, 64 which detect key bit patterns within the standard framing bytes for controlling the phases of the bit-distribution and byte out-latch clocks 41, 48. Additional comparator circuitry 34, 35, 36 employs framing byte sequences established during synchronous byte output to detect and signal the occurrence of frame structure benchmarks from which data-processing CMOS circuitry can determine the boundaries of data bytes within the parallel byte output from the demultiplexed frame. The phase-control bit sequence comparator circuitry 62, 64 is disabled during periods of satisfactory frame processing, but is reactivated upon the detection of framing sequence error to provide resynchronization in order to ensure recovery of properly restaged data bytes.

21 Claims, 8 Drawing Sheets

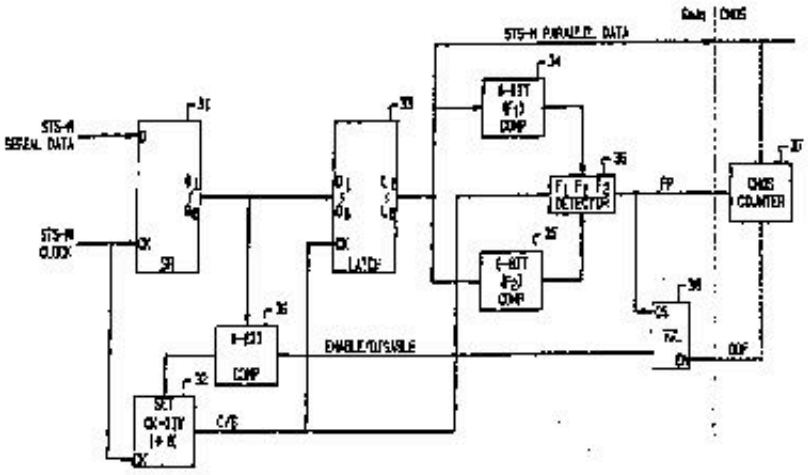


FIG. 1

FIG. 1

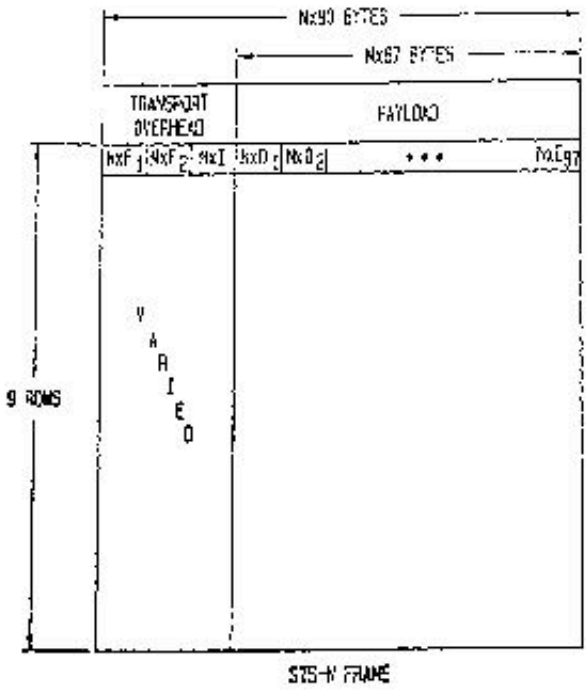


FIG. 2

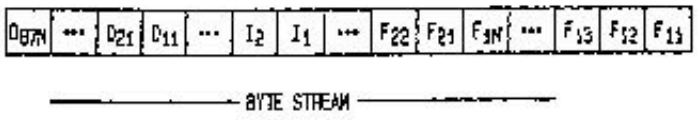


FIG. 2

FIG. 3

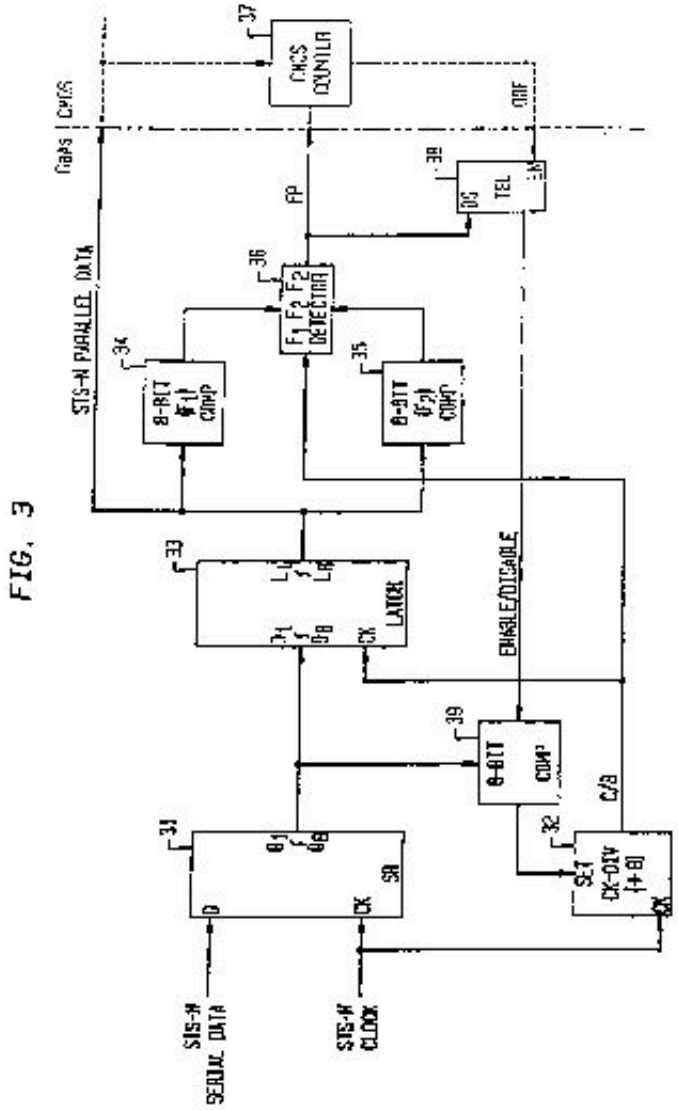

37
CMOS
GaAs
CMOS
OMF
38
FP
36
34
35
STS-N PARALLEL DATA
33
ENABLE/DISABLE
39
32
C/8
31
SDATA
SCLOCK

FIG. 3

FIG. 4

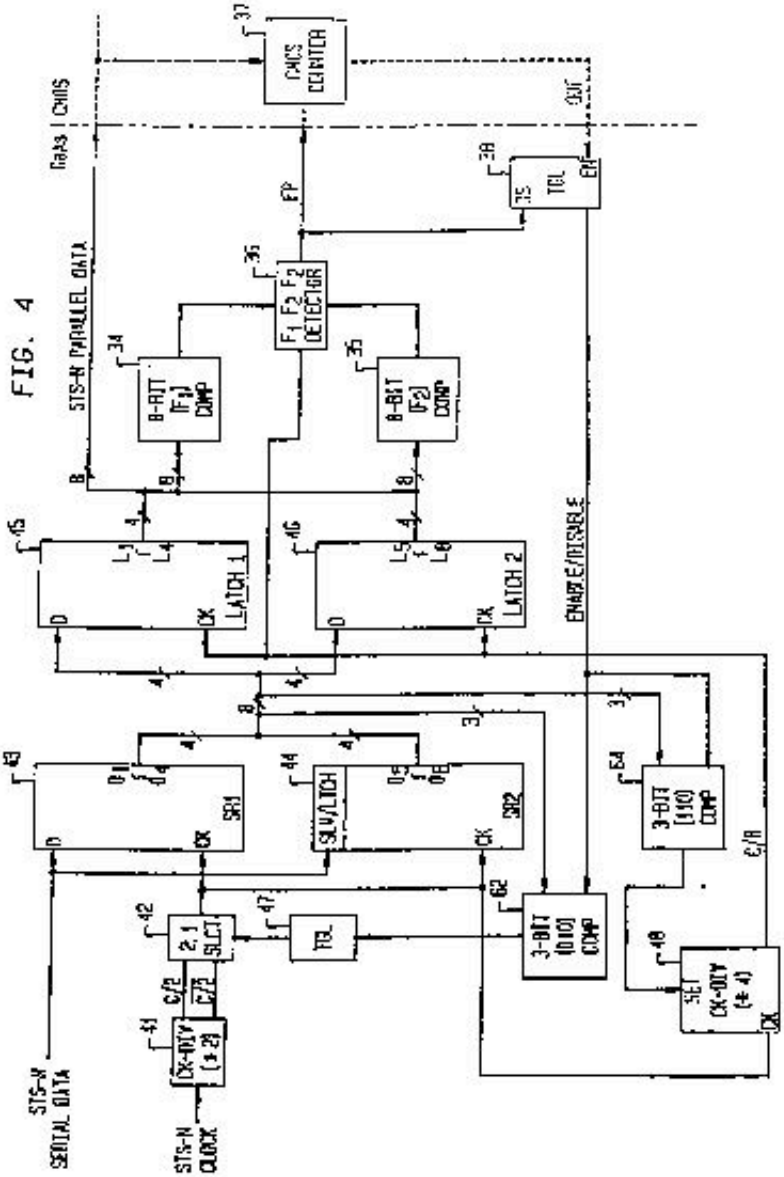


FIG. 4
STS-N PARALLEL DATA
8-BIT (F1) COMP
8-BIT (F2) COMP
F1 F2 DETECTOR
LATCH 1
LATCH 2
ENABLE/DISABLE
SLV/LATCH
3-BIT (110) COMP
3-BIT (010) COMP
2:1 SLCT
STS-N SERIAL DATA
STS-N CLOCK

FIG. 5

FIG. 5

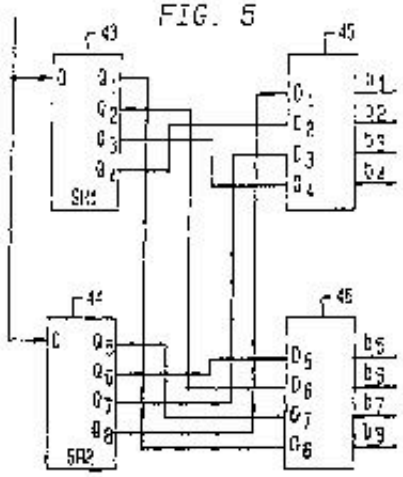


FIG. 6

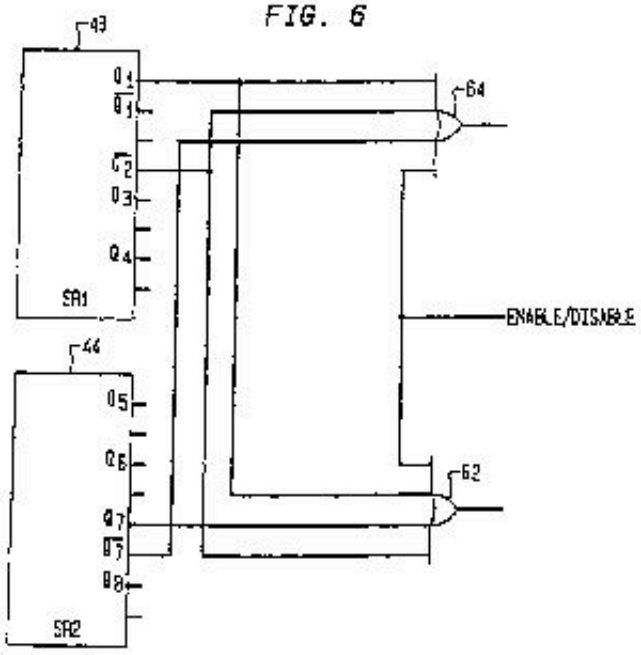
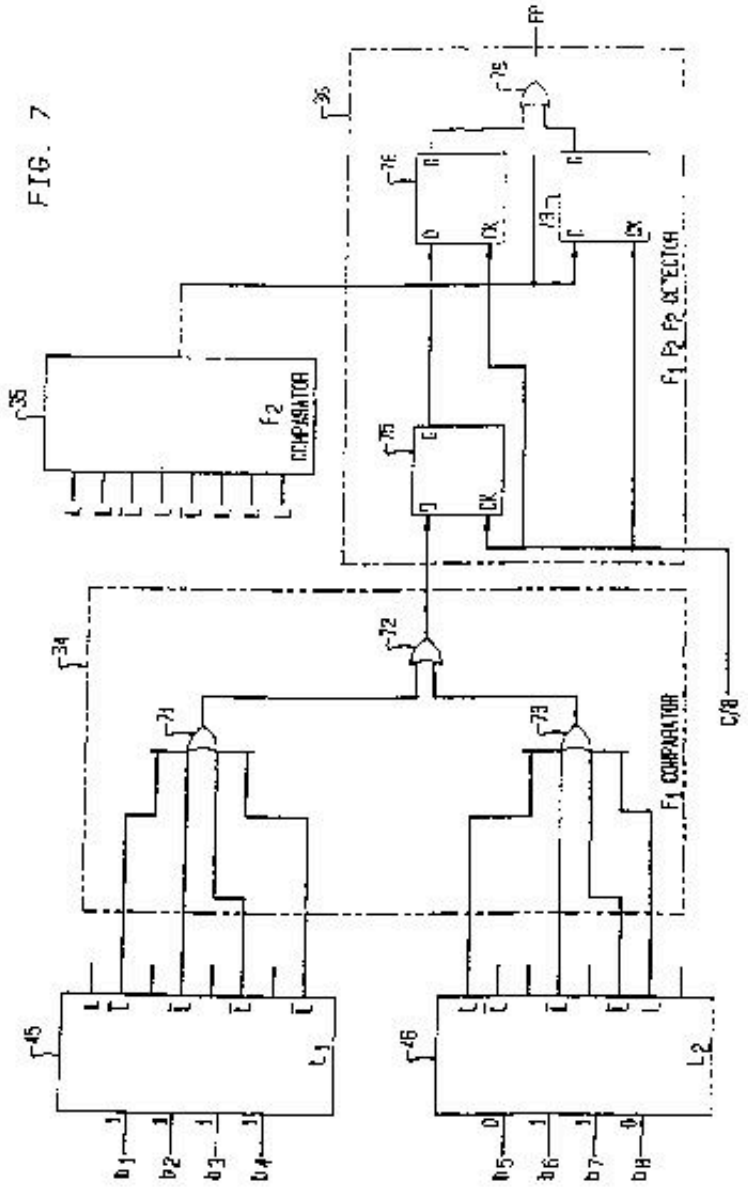


FIG. 6

FIG. 7

FIG. 7



*682

HIGH SPEED DIGITAL SIGNAL FRAMER-DEMULTIPLEXER

BACKGROUND OF THE INVENTION

The expansive data transmission capabilities of optical fiber technology have made practical the operation of digital telecommunication systems at rates well into the gigabit per second (Gbit/s) range. The advantages to be realized in this technology are apparent, and development of such systems has proceeded on numerous fronts worldwide. Unfortunately, these contemporaneous developments have resulted in a number of independently-devised signal architectures which lack the compatibility necessary for effective global, or even regional, communications networks.

With a view toward establishing and maintaining such compatibility, standards bodies have endorsed basic structures of optical system transmission rates and interfaces, not the least among which are those incorporated into the Synchronous Optical Network (SONET) hierarchy concept. This important advancement operates upon a base level digital signal framing format, namely the Synchronous Transport Signal level 1 (STS-1) frame, which consists of 810 8-bit bytes of data, and which therefore provides a serial bit transmission rate of 51.84 Mbit/s at the basic 8000 per second frame transmission rate.

Under this concept, signal transmissions of higher rate are achieved by interleaving bytes of any desired number of STS-1 frames in a prescribed sequence to form the correspondingly higher signal levels, e.g. STS-3, STS-4, STS-8, . . ., STS-24, etc. The STS-24 signal thus consists of the interleaved bytes of 24 STS-1 signals, and has a resulting transmission rate of 1244 Mbit/s, or 1.244 Gbit/s, i.e. 24 times the rate of the basic 51.84 Mbit/s of the STS-1 signal. For the transmission of such a signal, a multiplexed serial bit stream is assembled by interleaving repeated sequential extractions of one byte from each of the component STS-1 frames. It is necessary, therefore, that the signal receiver reconstruct from this serial bit stream the original base frame, or some frame multiple thereof, in order that the correct substance of the transmitted signal may be recovered.

While with current technology the serial bit stream may be assembled into fundamental 8-bit byte structures, it is essential to the proper recovery of the original SONET frames that the byte assembly be correctly synchronized and the boundaries of each such frame be identified in the bit stream transmission in order that the reconstructed bytes will duplicate each of the bytes which were interleaved to produce that serial transmission signal. The present invention provides method and apparatus to ensure that such proper synchronization and frame identification are established and maintained throughout such a signal transmission.

SUMMARY OF THE INVENTION

The basic SONET frame prescribed for the first transport level (STS-1) consists of nine rows of ninety 8-bit bytes each. Of these bytes, the first three in each row constitute the frame transport overhead containing framing, identification, error checking, and like information, while the remaining eighty-seven bytes make up the "payload" of the frame, i.e. the transport medium for the substance of the message or data transmission.

With a transmission rate of 51.84 Mbit/s, the STS-1 frame establishes the SONET frame period of 125 microseconds. This frame period is maintained throughout the hierarchy of increasing transport level frames by interleaving the respective bytes from each row of the component lower level frames, thereby deriving a transmission rate of N×51.84 Mbit/s for the STS-N frame. Utilizing available gallium arsenide (GaAs) enhancement-depletion mode metal semiconductor field effect transistor (MESFET) technology, integrated circuits for accomplishing such byte interleaf multiplexing have become practicable to the N=24 range of an STS-24 frame having a transmission rate of 1.244 Gbit/s.

Transmission of the STS-N frame is effected in a row-by-row manner, beginning with the first framing byte in the transport overhead and proceeding through the final N×87th payload byte of the first row before continuing on to the first overhead byte of the second frame row for transmission of each subsequent row of the frame in like manner. Following transmission of the last payload data byte of the ninth frame row at the end of the 125 microsecond frame period, the first framing byte of the next STS-N frame is transmitted, and the process continues in this manner throughout the transmission.

The bit stream of the transmission proceeds in the noted byte-interleaved succession at the rate, assuming the STS-24 frame, of 1.244 Gbit/s to the receiving station where that stream must be reformatted into the original bytes and frames in order for the receiver processing circuitry to properly extract the transmitted data and messages. Within this serial transmission of the data bit stream, however, there are no distinctly highlighted boundaries between the respective bytes and frames. It is necessary, therefore, that there be a capability in the receiving system by which these boundaries may be recognized so that synchronous byte structuring and frame formatting may be established.

The circuitry of the invention utilizes the two prescribed SONET framing byte bit patterns as bases for timing the initiation of byte structuring, as well as designating and confirming the boundaries of the frame format within such byte sequences. These framing bytes reside in the transport overhead and occupy the initial two positions in the STS-1 frame, or N-multiples thereof in a transmitted STS-N frame, and their respective unique bit patterns distinguish between them in all circumstances of bit pattern rotation.

During the demultiplexing of bits from the high-speed serial transmission of a frame, a characteristic bit pattern from one of the framing bytes is eventually recognized in comparator circuitry which signals the proper synchronization of byte formation and sets the clock controlling that operation. Other comparator means are provided which recognize the transition from the first to the second of the framing byte patterns to enable this occurrence to be utilized to denote the boundary between demultiplexed frames.

The present capabilities of GaAs enhancement-depletion mode and CMOS technologies are such as to provide maximum functionality of the former up to the STS-24 transmission rate of 1.244 Gbit/s, and of the latter at the 1:8 demultiplexed STS-3 rate of 155.5 Mbit/s. Although the reframing and demultiplexing of the high-speed serial data bit stream can be effected at the receiver in the GaAs MESFET circuitry, the requisite cost and power consumption make it desirable to reduce the signal transmission rate as soon as possible in the demultiplexing and signal processing operations in

order to take advantage of the more economical low power CMOS VLSI circuits. By limiting the operations of the high-speed GaAs circuits to the serial frame formatting and synchronization, and relegating to available CMOS chips the in-depth frame demultiplexing and signal processing, an effective and economical use is made of the capabilities of both these technologies.

In the general application of the present invention, the high-speed serial bit stream of the STS-N level, e.g. STS-24, transmission is demultiplexed to the basic SONET 8-bit byte-parallel format in the GaAs circuitry either at the STS-N clock rate or, preferably, at half that clock rate in order to provide a less restrictive time span for the implementation of the synchronizing gating functions. The resulting parallel byte stream is made available to companion CMOS circuitry for signal processing, while the high-speed GaAs chip, in addition to its byte-formatting function, is required only to recover framing synchronization. In the event of a loss of such frame synchronization, the CMOS processing circuit reinstitutes in the GaAs circuits the reframing process which normally will be accomplished within the period of two frames.

THE DRAWING

The present invention may be readily seen in the accompanying drawing of which:

FIG. 1 is the representation of an N-level frame of the Synchronous Optical Network (SONET) signal hierarchy;

FIG. 2 is the representation of the transmitted byte stream format of the first row of a SONET N-level frame;

FIG. 3 is a block diagram of an embodiment of the framer-demultiplexer circuit of the present invention;

FIG. 4 is a block diagram of a preferred embodiment of the framer-demultiplexer circuit of the present invention;

FIG. 5 is a block diagram of a shift register and latch arrangement utilized in the embodiment of the circuit of FIG. 4;

FIG. 6 is a block diagram of a shift register and 3-bit comparator arrangement utilized in the embodiment of the circuit of FIG. 4; and

FIG. 7 is a block diagram of an 8-bit comparator and frame boundary detector arrangement utilized in the framer-demultiplexer circuit of the present invention.

DESCRIPTION OF THE INVENTION

The Synchronous Optical Network (SONET) signal hierarchy is based upon the signal frame format generally represented in FIG. 1. The base, N = 1 signal frame of STS-1 (Synchronous Transport Signal level 1) consists of the nine rows of ninety bytes in which the first two bytes of 8-bits are the SONET framing bytes, F1 and F2, having the prescribed bit patterns, 11110110 and 00101000, respectively. The third byte of the first frame row, designated generally as 1, along with the remaining first three bytes in each of the remaining eight rows of the frame make up the balance of the transport overhead which provides frame identification, error checking information, message pointers, and the like.

The body of functional data, designated as the "payload", transmitted in each frame is located in the remaining 87 data bytes, D, in each of the nine frame rows to yield 781 bytes of such functional data. Each SONET frame is transmitted row-by-row at the rate of 8000 frames per second, thus producing, for the basic STS-1 signal, a serial bit stream of 51.84 Mbits/s. Successive levels of signal in the hierarchy are formed by interleaving the respective bytes of the basic STS-1 signals within the frame format to obtain the STS-N frame, where N = 2, 3, 4. . . . The basic 125 microsecond frame period is retained, however, thereby yielding increasing bit transmission rates in N×51.84 Mbits/s.

In each such frame, the similarly positioned bytes from each STS-1 signal are assembled sequentially in a string in the like position of that frame, thus locating a byte, Bj, from the jth position of the jth STS-1 frame at the Bij position in the STS-N frame. This SONET multiplexing arrangement may be seen from FIG. 2 in which there is depicted a representative serial transmission of the first row of an STS-N frame. The transmitted byte stream is headed by F1 framing bytes, F11, F12, . . . , F1N, from each of the N interleaved STS-1 frames, followed by the F2 framing bytes, F21, F22, . . . , and the remaining bytes of the row down to the final 87th data byte of the Nth STS-1 frame.

The following rows of the STS-N frame and subsequent such frames are similarly transmitted in the continuing serial bit stream to their destined terminating SONET receiver where the frames must be reformatted by reconstructing and demultiplexing the transmitted bytes in original order and sequence. It is necessary, however, in order to effect such byte and frame structuring that the beginning of each STS-N frame, as embedded in the serial transmission, be identified as such and that byte formation be synchronized with that benchmark.

As noted, the serial transmission rate of the SONET frame is determined by the number of STS-1 signals multiplexed into the STS-N frame. It is, of course, desirable that this transmission rate be as great as possible in order to best exploit the extensive bandwidth available in today's fiber optic transmission facilities. At present, GaAs enhancement-depletion mode MESFET technology provides the capability of multiplexing/demultiplexing SONET frames up to the STS-24 signal level of 1.244 Gbits/s. However, in practice, the signal receiver requires complex circuitry to carry out the overhead processing and payload extraction on incoming signals. It is for this reason that it is desirable for the maximum amount of receiver processing to be accomplished in widely available lower speed, low-power CMOS VLSI circuits in order to avoid the substantial cost and power requirements of high-speed GaAs chip processing.

In accordance with the present invention, the byte formatting and frame definition are accomplished in a high-speed GaAs MESFET device receiver arrangement generally shown in FIG. 3. In this embodiment, the transmitted serial data bit stream of the STS-N signal, which for purposes of this description will be assumed to be at the STS-24 level, is input to an 8-bit shift register (SR) 31 where it is clocked through to the Q-outputs at the STS-24 rate of 1.244 Gbit/s. These outputs of SR 31 are connected in parallel to the inputs of 8-bit latch 33 from which the data will be appear as 8-bit bytes at outputs L1 . . . L8. Although depicted here simply as separate devices, the shift register and latch may be combined in any known manner into a single device.

The STS-24 clock signal which is synchronized to the bit stream transmission, and which controls the sequencing of data bits through SR 31, is directed to clock divider 32 where it is reduced to one-eighth, C/8, of the

STS-24 rate, i.e. to 155.5 Mbit/s. This C/8 clock signal output from divider 32, with usual appropriate timing and delay adjustments, is input to latch 33 to thereby trigger the output of each byte of 8-bits newly accumulated in SR 31. Without further control, however, the byte formatting at this point is subject to the arbitrary phase of the counter of divider 32. The correct sequence of bits in any byte output from latch 33 can therefore not be assured, since, depending upon the set of the divider counter, the bits of such byte may be distributed in any fashion between two consecutive latched-out bytes. The known sequence of the prescribed F1 framing byte, 11110110, may, for example, appear in any of eight such distributions, such as xxxxx111, 10110xxx, or xx111101, 10xxxxxx.

In order to set the byte-latching clock signal from divider 32 to the proper phase to ensure synchronization of byte formatting with the original bytes of the frame, comparator 39 is used to monitor the progressing states of the outputs of SR 31 as the incoming serial data stream containing the bits of the F1 framing byte are shifted through. The 8-bit comparator 39, which may be an OR gate configuration such as that of F1 comparator 34 shown in FIG. 7, or any equivalent combination of other types of gate elements, such as AND gates (for example, is connected by means of individual input conductor leads to the appropriate Q and $\overline{Q}$ outputs of SR 31 in order that each such input will have a "0" state when SR 31 is loaded with the F1 framing byte, 11110110.

The "0" state output from comparator 39 will then set the counter of clock divider 32 to trigger the output of the matched F1 framing byte, and to begin clocking reconstructed, properly-phased 8-bit bytes out of latch 33 from that time on until some extraneous error occurs in the transmission. An enable/disable input to comparator 39, of which more will be described later, ensures that the resetting of clock divider 32 takes place only when its counter is out of synchronization with the F1 framing bytes.

Upon completion of the formatting of the high-speed input serial data stream to a low-speed, properly synchronized byte-parallel data stream, there remains the problem of identifying the boundaries of each frame of the original transmission in order that the payload, as well as the relevant overhead information bytes, may be demultiplexed to the basic STS-1 level. For this purpose, the present invention relies upon the prescribed bit sequences of both the F1 and F2 framing bytes, of which each frame above STS-1 will have at least two, the F2 bytes following immediately upon the final F1 framing byte as depicted in FIG. 2.

This transition from the F1 to the F2 framing bytes repeats once each frame, N bytes after the beginning of the frame, and therefore serves as the benchmark from which may be determined the boundaries of the frame to be processed in the receiver circuitry. To recognize this transition, occurrence of the unique byte pattern sequence, F1F2F2, that is 11110110, 00101000, 00101000, is detected in the combination of 8-bit comparators 34, 35, and F1F2F2 detector 36, an embodiment of which is shown in greater detail in FIG. 7.

As earlier noted, the inputs to the OR gate combination of F1 comparator 34 are attached to those respective L or $\overline{L}$ outputs of 8-bit latch 33, or of the two 4-bit latches 45, 46 used in the embodiment of FIG. 4, which will present "0" states to each of OR gates 71, 73 when an F1 framing byte, 11110110, is latched to the byte-parallel output line in synchrony with the C/8 clock signal from divider 32. The like "0" state outputs from gates 71, 73 will then carry through gate 72 as the output from F1 comparator 34. Passing sequentially through flip-flop (F/F) devices 75, 76, this "0" state output will appear at OR gate 79 two C/8 clock pulses later.

In similar manner, a following F2 framing byte, 00101000, will appear at comparator 35, which has the same device component structure, but different input lead arrangement, as comparator 34, one C/8 clock signal state change, or pulse, after the F1 byte appearance at comparator 34, and will provide a "0" state output to F/F 78. This state will appear at gate 79 one clock pulse later along with the second "0" state from comparator 35 signifying the occurrence of the second F2 byte in the F1F2F2 sequence. The three simultaneous "0" states thus appearing at gate 79 from comparators 35 and F/Fs 76 and 78 confirm detection of the unique F1F2F2 byte sequence by output of a framing pulse, FP, from detector 36.

This framing pulse, FP, will occur once in each synchronous frame of the byte-parallel transmission and is sent from the GaAs chip to a byte counter 37 associated with processing CMOS circuitry to reset that counter when the transmission remains "in frame". In the event of any error which causes loss of byte synchronization, F1 and F2 comparisons and F1F2F2 detection will fail, resulting in loss of the framing pulse, FP. After two frame cycles of such an "out-of-frame" condition, the CMOS byte counter 37 will have accumulated a preselected count and will overflow an out-of-frame pulse, OOF, which is directed back to toggle 38 in the GaAs circuitry to create an "enable" condition in comparator 39.

Thus activated, comparator 39 will initiate the reframing procedure with a search of SR 31 output conditions until the F1 framing byte appears. The phase of divider 32 is thereupon reset to establish, once again, correct frame synchronization. The first F1F2F2 sequence detected thereafter will generate an initial framing pulse, FP, which, in addition to resetting CMOS counter 37, will trip toggle 38 to disable comparator 39 and allow divider 32 to remain set at its present phase for as long as the transmission remains in frame.

The foregoing embodiment, although effective in its implementation, does exhibit somewhat less than optimal performance in that the initial operations for resetting the framing clock are carried out at the STS-N clock rate. Where, as in the current example, transmission is at the STS-24 signal level, these operations not only require the use of excessive power in the necessary high-speed devices, but they also establish a rather restricted clock-setting "window" which extends for only about 800 picoseconds. In order to effect an improvement in these areas, the split-register embodiment of the invention depicted in FIG. 4 is preferred over the single-register implementation of FIG. 3 in that it utilizes high-speed devices only in a simple free-running clock divider, and it expands the framing clock window to a more comfortable 1.6 nanoseconds.

As shown in FIG. 4, this preferred embodiment of the invention employs a high-speed clock divider 41 which need only reduce the STS-N, i.e. STS-24, clock rate of 1.244 Gbit/s to one-half that rate. Utilizing both the Q and $\overline{Q}$ outputs of divider 41, there are made available two 622 Mbit/s clock signals, C/2 and $\overline{C/2}$ that are in "180° phase opposition. One or the other of these clock signals may be put into use by means of 2:1 selector 42

which is set by toggle 47. This toggle is in turn controlled by 3-bit comparator 62 and activated when it is determined, as will later be described, that the byte formatting is out of frame and in need of the opposite phase of the C/2 clock. The selected clock signal is employed to load and shift the STS-24 signal serial data in paired bits into and through the two shift registers 43 and 44, and serves also as a basis for four-fold rate reduction in clock divider 48 to obtain the 155.5 Mbit/s byte clock signal.

Shift registers 43 (SR1) and 44 (SR2) are basically constructed of master/slave-type flip-flop devices which load input data on one phase, e.g. the rise, of the triggering clock, and latch that loaded data to the Q outputs on the opposite, i.e. falling, phase. Register 44, however, comprises as an additional first element a slave latch which operates, in this example, on the falling clock phase to trap a current bit for use as input to the following first stage of that register on the next rising load phase of the clock. In this manner, the trapped bit and the following bit in the serial transmission are loaded as a pair into the respective first stages of the SR2 and SR1 registers at each pulse of the C/2 clock, the SR2 bit lagging the SR1 bit due to the delay imparted by the trap latch element. Thus, although this clock signal is running at only half the rate of the serial data transmission, each STS-24 data bit is nonetheless clocked into the respective registers.

Latching out of the eight bits accumulated in registers 43, 44 is effected, as in the previous embodiment, upon a C/8 clock signal derived from the STS-24 clock of the incoming serial data stream. In this instance, this latching clock signal is obtained from a four-fold division, in clock divider 48, of the C/2 signal from selector 42. It should be understood here that although there are depicted a pair of 4-bit latches 45, 46 in use for this purpose a single 8-bit latch might be employed as in the single stage embodiment of FIG. 3. In any event, in order to obtain the correct sequence of bits at the latch outputs L1 . . . L8, the arrangement of conductor leads between the Q1 . . . Q8 outputs of registers 43, 44 and the inputs to latches 45, 46 is selected to be as shown in FIG. 5.

Since the first bits input to registers 43, 44 will have shifted to their respective Q4 and Q8 output positions during the accumulation of the remaining six bits of a given byte, the lead pattern between registers 43, 44 and latches 45, 46 appears as Q8-D1, Q4-D2, Q7-D3, . . . This chosen arrangement will, of course, be valid for only one of the two possible opposed clock phases deriving from selector 42; however, as noted, the clock phase may readily be reversed to match the indicated connector arrangement. Upon each pulse of a properly-phased C/8 clock signal, then, the eight bits accumulated in registers 43, 44 in frame synchronization will be latched out to the byte-parallel output line in a correctly ordered, i.e. b1 . . . b8, byte.

As previously indicated, the extra trap latch stage in register 44 imparts a one-bit delay in the loading of its first stage, thereby causing the SR2 bit to lag its companion SR1 bit during each clocked step in the register-loading process. As a result, the first bit of a given byte will, depending upon the phase of the C/2 clock, be loaded into SR1 register 43 or SR2 register 44. In the former event, the lagging SR2 bit will be the last bit of the previous byte, and the ultimately loaded byte will be out of byte synchronization. In the latter condition, the SR2 bit, i.e. the first bit of the loading byte, will lag the more recently arrived second bit of that byte which will be loaded simultaneously into the first stage of register 43 as the SR1 bit, thereby establishing the byte-synchronous condition wherein all bits of the given byte will reside in the registers at one time during the loading progression. Thus, in the present example, the byte-synchronous condition exists when the odd bits, i.e. the first, third, fifth, and seventh, of the given frame are trapped at SR2 register 44, and the even bits are loaded into SR1 register 43.

Since the beginning of a framing procedure is subject to an arbitrary phase of the STS-N clock, and its dependent loading clock signal, C/2, one cannot be assured of the accumulation of bits in a byte-synchronous fashion, i.e. synchronized in accordance with the above-described loading preference upon which the noted conductor arrangement was chosen for transmitting staged bits to the byte-parallel output latch(es). There is thus a 50% chance that first bit of the 8-bit sequence of a given byte will be input to register 43 on the rising clock, rather than being, as desired, trapped at the slave latch of register 44 on the falling clock pulse. The clocking of bits in this out-of-phase manner will in effect retard the formatting of the byte by one bit and cause the latched-out byte to be out of phase, with the resultant loss of frame as well as all substantial meaning of the content of the transmission. The F1 framing byte, for instance, assuming a properly phased latching clock, C/8, would not appear in a fully-loaded register pair as its prescribed [1]1[0]10, but as an out-of-phase pattern, x[1]1[0]11.

Upon analysis it will be seen, however, that during the progressive loading of registers 43, 44 under an out-of-phase clock signal a unique pattern of bits from an F1 framing byte will appear at the Q-outputs of those registers; specifically, the 010 pattern will appear at the Q7, Q2, Q1 outputs. The unique character of this pattern lies in the fact that it will not thus appear during any progressive loading of an in-phase sequence of any number, i.e. from any STS-N frame, of F1 and F2 framing bytes. The appearance of the 010 pattern may be relied upon, therefore, to signal the existence of an out-of-phase loading clock signal at the beginning of a framing procedure, since it will be encountered during receipt of the first F1 framing byte of the STS-N frame, and may be used to trigger a change in the output of selector 42 to the phase-opposed C/2 loading clock signal.

This phase-change operation is controlled in 3-bit (010) comparator 61, which is shown in FIG. 6 as being implemented in OR gate 62 to output a "0" state which will activate toggle 47 to effect the clock phase change in selector 42, as previously noted. To achieve the required 000 input to gate 62, the Q7, $\overline{Q2}$, Q1 outputs from registers 43, 44 are employed, along with the "0" input "enable" state from toggle 38 which, as will be recalled, is in that state as a result of a failure of the detection of the F1F2F2 frame sequence, i.e. the indication of the existence of some out-of-frame condition. As a matter of convention, the three bit leads are simply indicated in FIG. 4 by the character "/3", in the same manner as the respective leads for 4-bit and 8-bit data lines bear the notations, "/4" and "/8". Once the loading clock signal has been thus set to the proper phase, comparator 62 will not again see the occurrence of the 010 bit pattern in subsequent F1 and F2 framing bytes, and will be disabled at the F1F2F2 transition; thereby allowing the clock signal to remain in the selected phase even in the

*686

event that bits of an actual data byte match the 010 pattern. If, however, a transmission error occurs which disrupts the established framing, comparator 62 will be re-enabled by the resulting F1F2F2 failure at the beginning of the next frame, and will again initiate proper loading clock phasing at the start of the following frame.

With the loading clock signal, C/2, in the correct phase to ensure the loading of proper bytes from the STS-N frame, there remains the necessity to set the latching clock signal, C/8, to the correct phase to properly formatted bytes, rather than some intermediate rotation or progression in the bit accumulators. Once again, an analysis of the progressing in-phase bit patterns at the Q-outputs of registers 43, 44 reveals that there appears at the noted Q7, Q2, Q1 outputs the 110 bit pattern only when a complete F1 framing byte is fully-loaded and ready to be latched to the 3-bit byte-parallel output line. In the manner previously described with respect to comparator 62, a second 3-bit (110) comparator 64, more specifically shown in OR gate implementation in FIG. 6, employs the outputs from $\overline{Q7}$, $\overline{Q1}$ and Q1 to obtain, with the enabling state from toggle 38, to set the counter of clock divider 48 to trigger at this byte-synchronized stage for all subsequent framed bytes in the transmission. The disabling and re-enabling of comparator 64 is effected in the same manner as, and coincides with, that of comparator 62.

After frame synchronization has been established in the foregoing manner for the preferred embodiment of FIG. 4, the procedure for frame boundary definition is carried out as previously described with respect to the single-register embodiment of FIG. 2, namely by passing the synchronous framing bytes on to 8-bit comparators 34, 35 to initiate the confirming framing pulse, FP, from F1F2F2 detector 36 during the subsequent in-frame condition. The enable/disable signal from toggle 38, which is conditioned by the framing pulse, is likewise employed similarly in both embodiments to activate the clock-phasing comparators. Although not specifically shown in the drawing, this signal may be used in comparator 39 in the same manner as that employed with 3-bit comparators 62, 64 (FIG. 6) to supply the additional control input state. For example, this control signal may be input to a final stage of 8-bit comparator 39, which could be similar to that shown as a gate 72 in comparator 34 (FIG. 7).

The present invention thus provides for the maximum utilization of available technologies for optimum economies of power and time in the demultiplexing of high-speed serial bit data transmissions to low-speed byte-parallel format within the Synchronous Optical Network (SONET) signal hierarchy. It is anticipated that other embodiments of the invention will be apparent from the foregoing description to those of ordinary skill in the art, and such embodiments are likewise to be considered within the scope of the invention as set out in the appended claims.

What is claimed is:

1. Apparatus for demultiplexing a serial data bit stream consisting of a continuum of an interleaved multiplicity of data bytes of predetermined size derived from a plurality of identically-formatted contributory frames each containing a plurality of said data bytes, and for reconstructing said data bytes and identifying from among them a benchmark from which may be determined the beginning byte of each of such contributory frames and, thereby, the boundaries of such frames, said apparatus comprising:

(a) means for accumulating data bits from said serial stream to form bytes having the same predetermined number of bits as do said interleaved data bytes;

(b) means for comparing at least one bit pattern from each byte thus formed with at least one bit pattern known to have comprised a byte of each of said contributory frames and for providing a first signal when a match is detected between said compared patterns;

(c) means responsive to said first signal for effecting the output of the byte of matching bit pattern, and each byte thereafter formed of newly accumulated bits, as said reconstructed bytes;

(d) means for comparing bit patterns from a contiguous plurality of said output reconstructed bytes with bit patterns known to have comprised a like contiguous plurality of bytes of each of said contributory frames, and for providing a second signal as said benchmark identification when a match is detected between said compared patterns; and

(e) means responsive to said second signal for controlling the operability of said means for providing said first signal.

2. Apparatus according to claim 1 wherein said data bit accumulating means comprises:

(a) shift register means providing a total number of stages equal to said predetermined number of bits; and

(b) clock means providing a signal comprising states for loading said data bits into said register means and shifting said bits through said stages.

3. Apparatus according to claim 2 wherein said register means comprises a single shift register incorporating said total number of stages, and said clock means is arranged to provide said signal states at the rate of the transmission of said serial data bit stream.

4. Apparatus according to claim 2 wherein said register means comprises a plurality of shift registers of which each incorporates the same number of stages, and said clock means is arranged to provide said signal states at a rate which is obtained by dividing said bit stream transmission rate by the number equal to said plurality of shift registers.

5. Apparatus according to claim 4 wherein said register means comprises a pair of shift registers and means for presenting the individual data bits of consecutive bit pairs in said serial stream at the respective register inputs during the register-loading states of said clock signal.

6. Apparatus according to claim 5 wherein said means for presenting data bits comprises latch means associated with one of the registers of said pair for trapping and presenting at the input of said one register the individual data bits in said serial data stream which appear at said latch means during the states of said clock signal that are phase-opposed to said register-loading states.

7. Apparatus according to claim 6 wherein said clock means is arranged to selectively reverse the phase of said clock signal.

8. Apparatus according to claim 7 wherein said clock means is arranged to provide a pair of clock signals in phase opposition, and comprises means for selecting one of the clock signals of said pair.

9. Apparatus according to claim 7 which further comprises:

(a) means for comparing at least one bit pattern from each byte formed by the accumulation of data bits at the output of said shift register pair with at least one bit pattern known to appear in such a byte formed from a known contributory frame byte only during a given one of the available phases of said clock signal;

(b) means for providing a third signal when a match is detected between said compared patterns;

(c) means responsive to said third signal for effecting the selective reversal of said clock signal phase; and

(d) means responsive to said second signal for controlling the operability of said means for providing said third signal.

10. Apparatus according to claim 1 wherein said bit pattern comparing and first signal providing means comprises gate means the inputs to which are derived from selected outputs of said bit accumulating means, which output selections are such, in location and signal state, as to input to said gate means a contributing activating state for each bit of said known bit pattern that appears at said outputs.

11. Apparatus according to claim 1 wherein said bit pattern comparing and second signal providing means comprises:

(a) a plurality of first gate means the inputs to which are derived from selected outputs of said reconstructed byte output means, which output selections are such, in location and signal state, as to input to respective ones of said first gate means a contributing activating state for each bit of said known bit patterns of respective ones of said contiguous contributory frame bytes that appears at said outputs;

(b) second gate means the inputs to which comprise the outputs from said first gate means; and

(c) means for delaying the signals output from said respective first gate means so as to effect the simultaneous appearance of said output signals at said second gate means inputs, thereby effecting output of said second signal.

12. Apparatus according to claim 11 for demultiplexing a serial data bit stream derived from the format of contributory frames which comprises a pair of contiguous bytes of different, known bit patterns, thereby establishing in the output reconstructed bytes the contiguous plurality of bytes consisting of one byte having the first and the following two bytes having the second of said pair of bit patterns, wherein said first gate means comprises a pair of gate arrangements the respective inputs to which are derived from the outputs of said reconstructed byte output means which represent the bit patterns of said pair of contiguous bytes.

13. The method for demultiplexing a serial data bit stream consisting of a continuum of an interleaved multiplicity of data bytes of predetermined size derived from a plurality of identically-formatted contributory frames each containing a plurality of said data bytes, and for reconstructing said data bytes and identifying from among them a benchmark from which may be determined the beginning byte of each of such contributory frames and, thereby, the boundaries of such frames, said method comprising:

(a) accumulating data bits from said serial stream to form bytes having the same predetermined number of bits as do said interleaved data bytes;

(b) comparing at least one bit pattern from each byte thus formed with at least one bit pattern known to have comprised a byte of each of said contributory frames;

(c) providing a first signal when a match is detected between said compared patterns;

(d) effecting in response to said first signal the output of the byte of matching bit pattern, and each byte thereafter formed of newly accumulated bits, as said reconstructed bytes;

(e) comparing bit patterns from a contiguous plurality of said output reconstructed bytes with bit patterns known to have comprised a like contiguous plurality of bytes of each of said contributory frames;

(f) providing a second signal as said benchmark identification when a match is detected between said compared patterns; and

(g) effecting in response to said second signal discontinuation of the provision of said first signal.

14. The method according to claim 13 wherein said data bit accumulating comprises:

(a) providing shift register means comprising a total number of stages equal to said predetermined number of bits; and

(b) loading said data bits into said register means and shifting said bits through said stages in response to a given clock signal.

15. The method according to claim 14 wherein said data bits are loaded at the rate of transmission of said serial bit stream into a single shift register incorporation said total number of stages.

16. The method according to claim 14 wherein said data bits are loaded into a plurality of shift registers, of which each incorporates the same number of stages, at a rate which is obtained by dividing said bit stream transmission rate by the number equal to said plurality of shift registers.

17. The method according to claim 16 wherein said data bits are loaded into a pair of shift registers by presenting the individual data bits of consecutive bit pairs in said serial stream at the respective register inputs during the register-loading states of said clock signal.

18. The method according to claim 17 wherein said data bits are loaded into one of said pair of shift registers by trapping in latch means associated with said one register said presenting at the input of said one register the individual data bits in said serial data stream which appear at said latch means during the states of said clock signal that are phase-opposed to said register-loading states.

19. The method according to claim 18 which further comprises:

(a) comparing at least one bit pattern from each byte formed by the accumulation of data bits at the output of said shift register pair with at least one bit pattern known to appear in such a byte formed from a known contributory frame byte only during a given one of two opposed phases of said clock signal;

(b) providing a third signal when a match is detected between said compared patterns;

(c) effecting in responsive to said third signal the reversal of the phase of said clock signal; and

(d) effecting in response to said second signal discontinuation of the provision of said third signal.

20. The method according to claim 13 wherein said comparing of bit patterns from a contiguous plurality of

*688

output reconstructed bytes, and said providing of said second signal comprises:

(a) inputting to respective ones of a plurality of first gate means a contributing activating signal state for each bit in an output reconstructed byte that matches, in state and position, a bit in said known bit patterns of respective ones of said contiguous contributory frame bytes;

(b) inputting to second gate means the outputs from said first gate means; and

(c) delaying the signals output from said respective first gate means so as to effect the simultaneous appearance of said output signals at said second gate means inputs, thereby effecting output of said second signal.

21. The method according to claim 20 for demultiplexing a serial data bit stream derived from the format of contributory frames which comprises a pair of contiguous bytes of different, known bit patterns, thereby establishing in the output reconstructed bytes the contiguous plurality of bytes consisting of one byte having the first and the following two bytes having the second of said pair of bit patterns, wherein said inputting of signal states comprises inputting to each respective one of a pair of gate arrangements a contributing activating signal state for each bit in an output reconstructed byte that matches, in state and position, a bit in said known bit patterns of the respective ones of said pair of contiguous contributory frame bytes.

* * * * *