Exhibit A

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.** **370/94.2**; 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	Schwärtz et al.	370/89
4,516,240	5/1985	Kume et al.	370/94
4,594,708	6/1986	Servel et al.	370/94
4,685,105	8/1987	Shikama et al.	370/89
4,706,246	11/1987	Kume	370/89
4,763,319	8/1988	Rozenblit	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. Muise, et al., "Experiments in Wideband Packet Technology", Proc. 1986, International Zurich Seminar on Digital Communications, pp. 136–138.

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

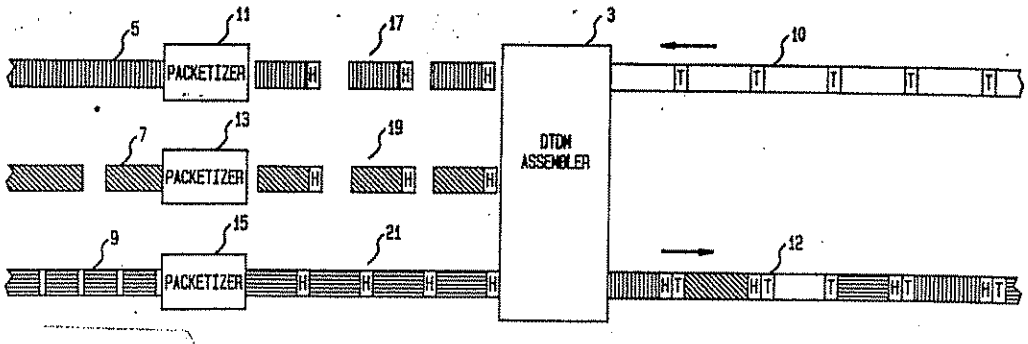
A. Thomas, et al., "Asynchronous Time Division Techniques: An Experimental Packet Network Integrating Video Communication", Proc. International Switching Symposium, May 1984.

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



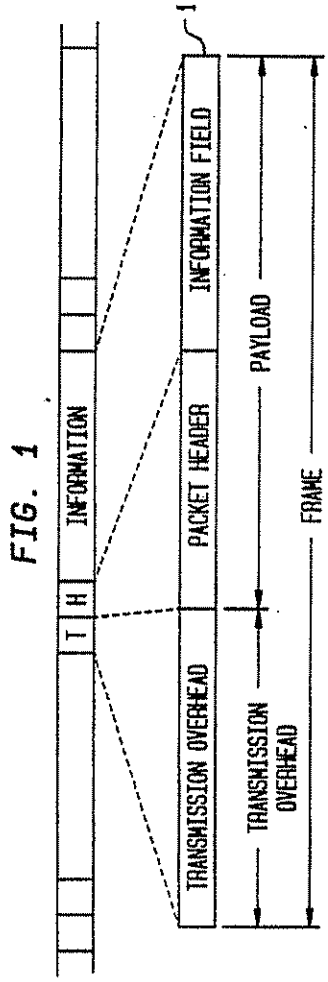

1
INFORMATION
T
H
INFORMATION FIELD
PACKET HEADER
TRANSMISSION OVERHEAD
PAYLOAD
TRANSMISSION OVERHEAD
FRAME

FIG. 1

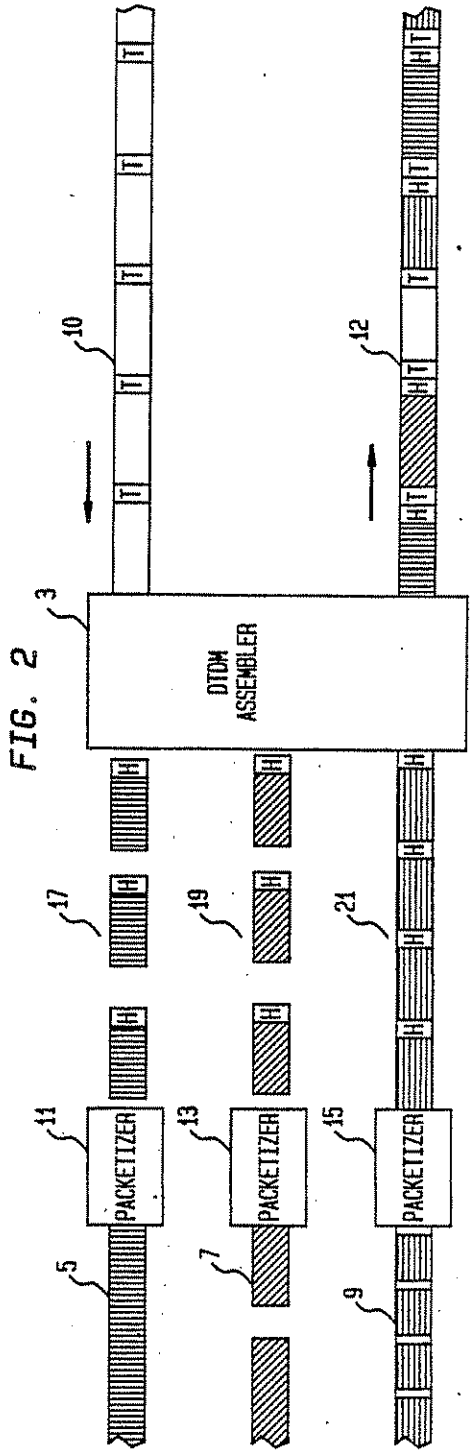
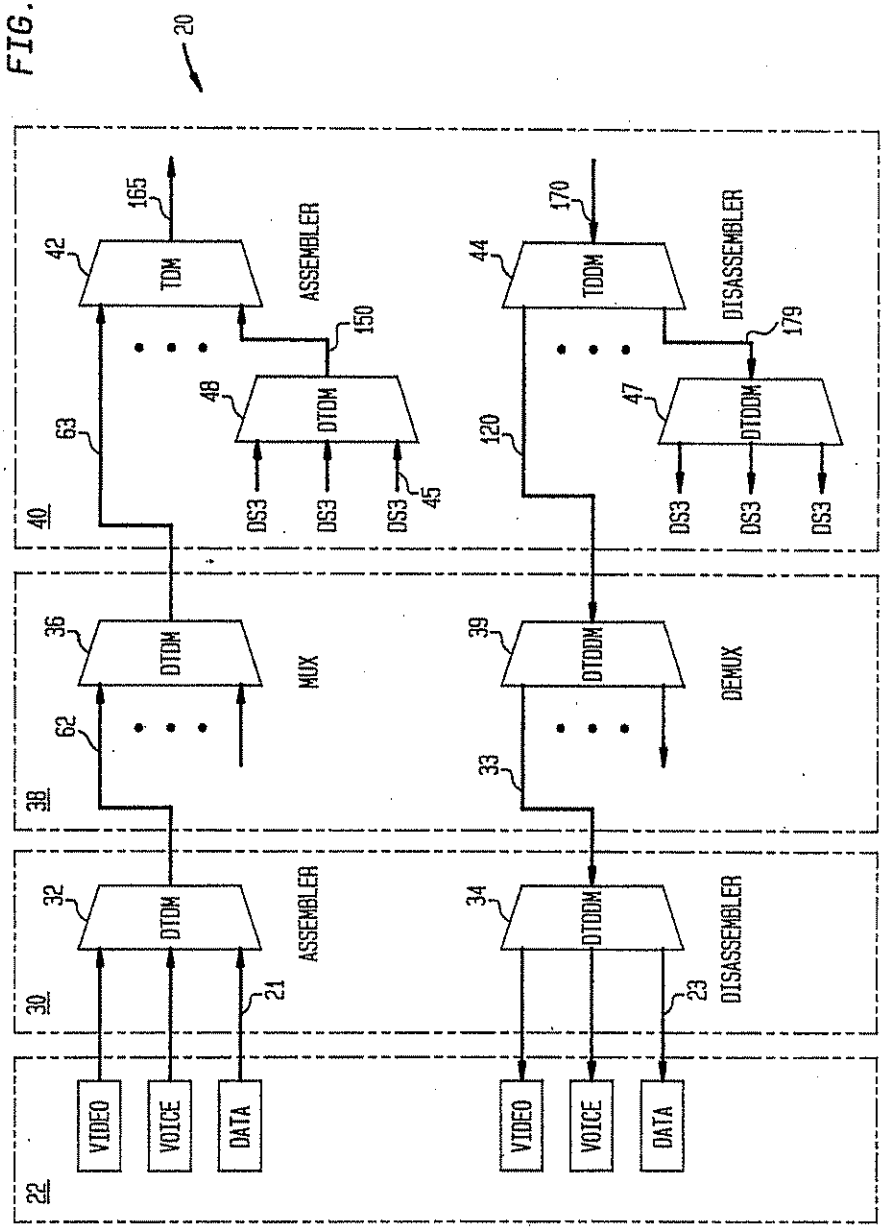

10
12
3
DTDM ASSEMBLER
17
19
21
11
13
15
PACKETIZER
PACKETIZER
PACKETIZER
5
7
9

FIG. 2

FIG. 3


20
165
42
TDM
ASSEMBLER
150
48
DTDM
63
DS3
DS3
DS3
45
40
170
44
TDDM
DISASSEMBLER
179
47
DTDDM
120
DS3
DS3
DS3
36
DTDM
MUX
62
38
39
DTDDM
DEMUX
33
32
DTDM
ASSEMBLER
21
30
34
DTDDM
DISASSEMBLER
23
VIDEO
VOICE
DATA
VIDEO
VOICE
DATA
22

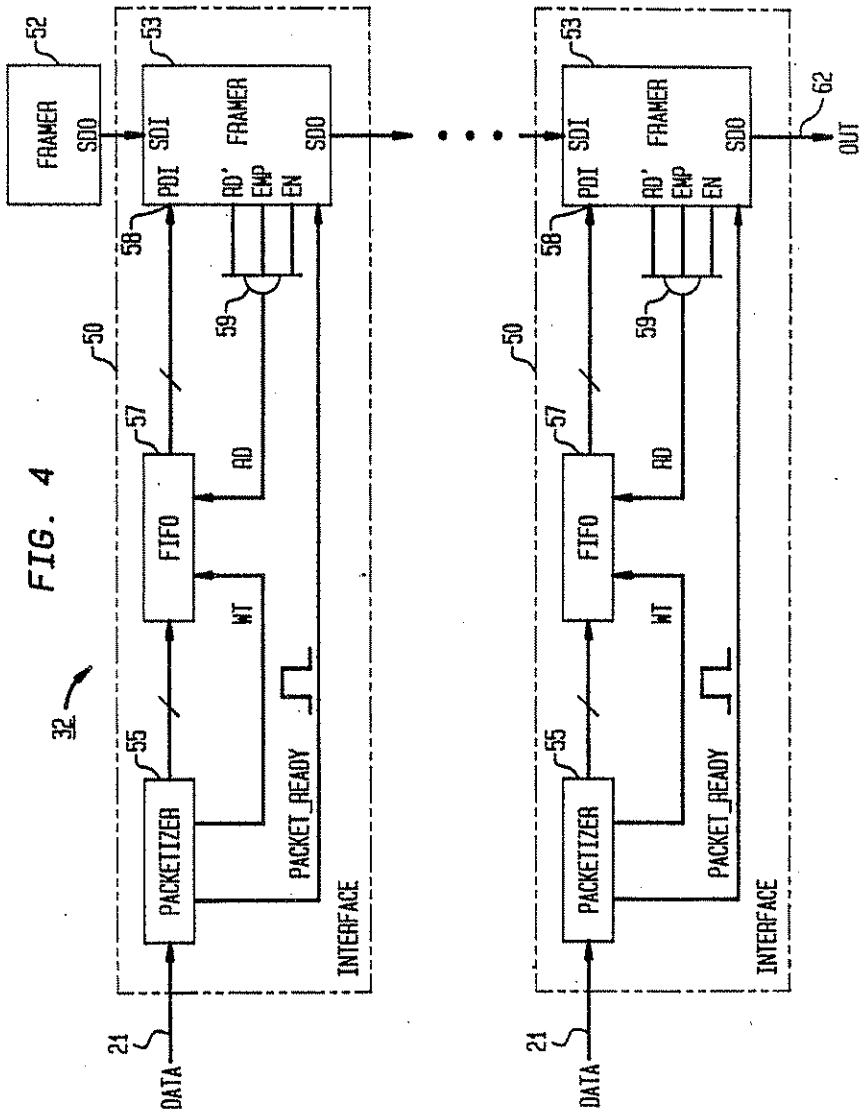
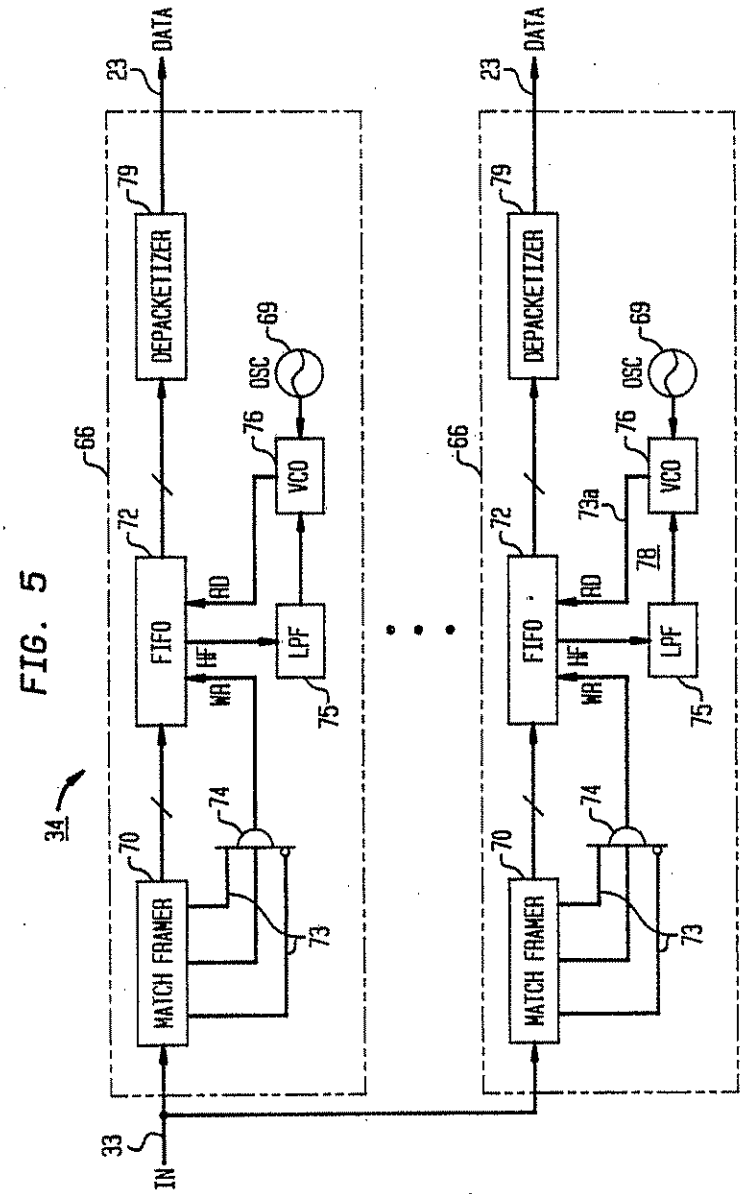

FRAMER
SDO
SDI
FRAMER
PDI
AD'
EMP
EN
SDO
OUT
FIFO
AD
WT
PACKETIZER
PACKET_READY
INTERFACE
DATA
52
53
58
59
50
57
55
21
62
32

FIG. 4

FIG. 5


34
33
IN
23
DATA
79
DEPACKETIZER
69
OSC
76
VCO
66
72
FIFO
AD
HF
WR
LPF
75
74
70
MATCH FRAMER
73
73a
78

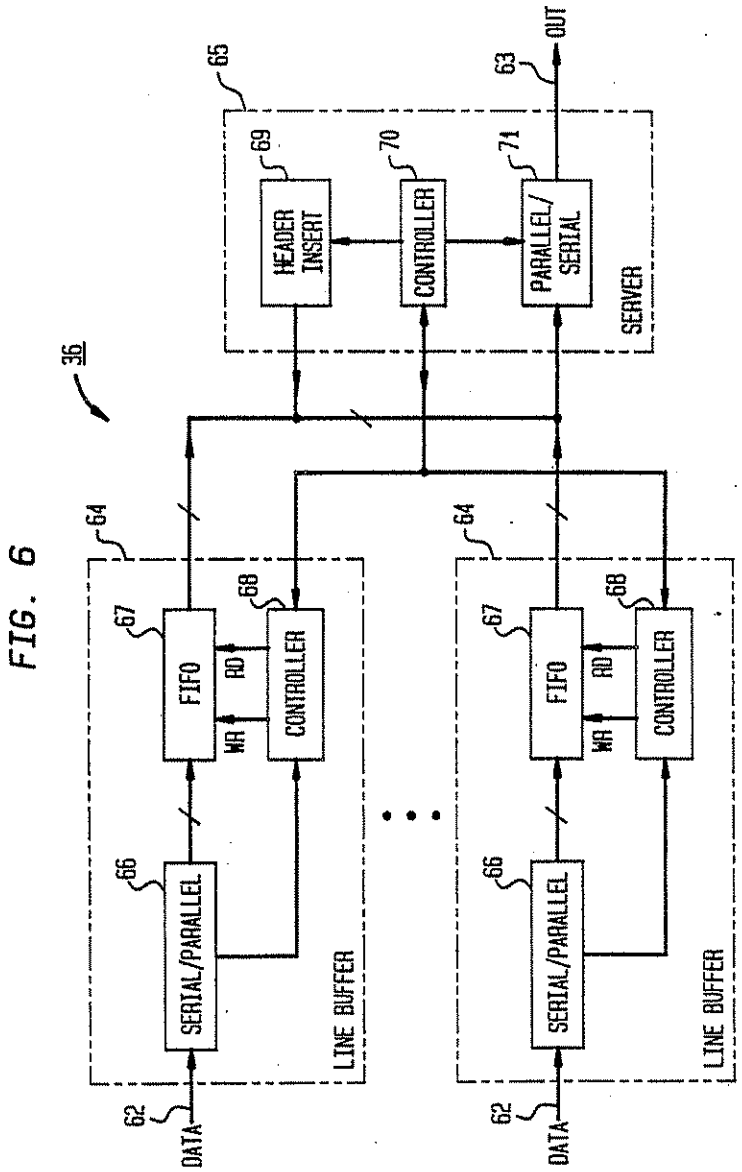

36
65
63
OUT
69
70
71
HEADER INSERT
CONTROLLER
PARALLEL/ SERIAL
SERVER
64
64
67
68
FIFO
RD
WR
CONTROLLER
66
SERIAL/PARALLEL
LINE BUFFER
62
DATA

FIG. 6

FIG. 7

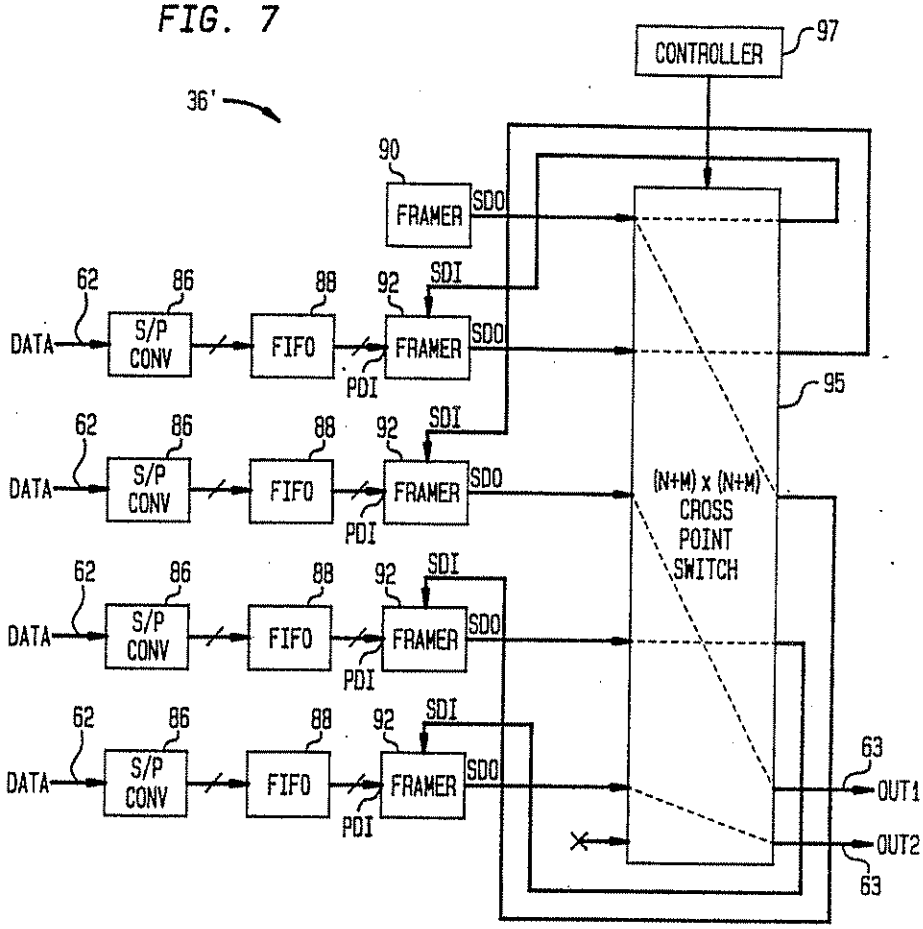

CONTROLLER
97
36'
90
FRAMER
SDO
SDI
62
86
88
92
DATA
S/P CONV
FIFO
FRAMER
PDI
95
(N+M) x (N+M) CROSS POINT SWITCH
63
OUT1
OUT2

FIG. 8

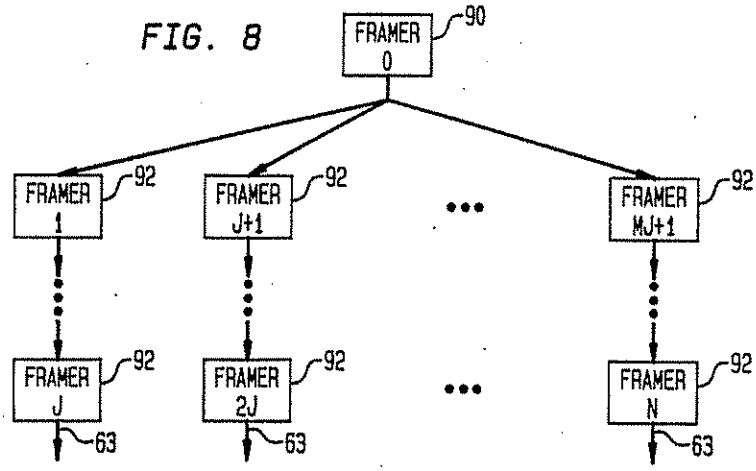
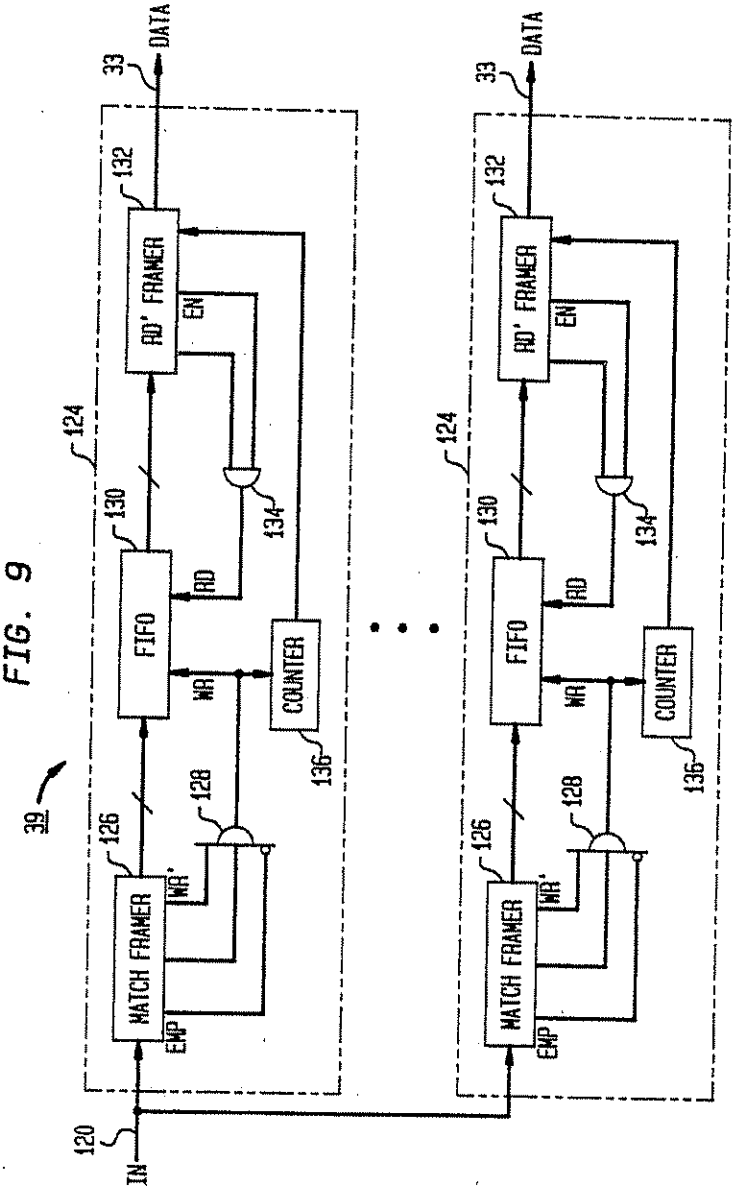

FRAMER 0
90
FRAMER 1
FRAMER J+1
FRAMER MJ+1
FRAMER J
FRAMER 2J
FRAMER N
92
63

FIG. 9


39
33
DATA
132
RD' FRAMER
EN
124
130
FIFO
RD
134
WR
COUNTER
136
128
126
WR'
MATCH FRAMER
EMP
120
IN

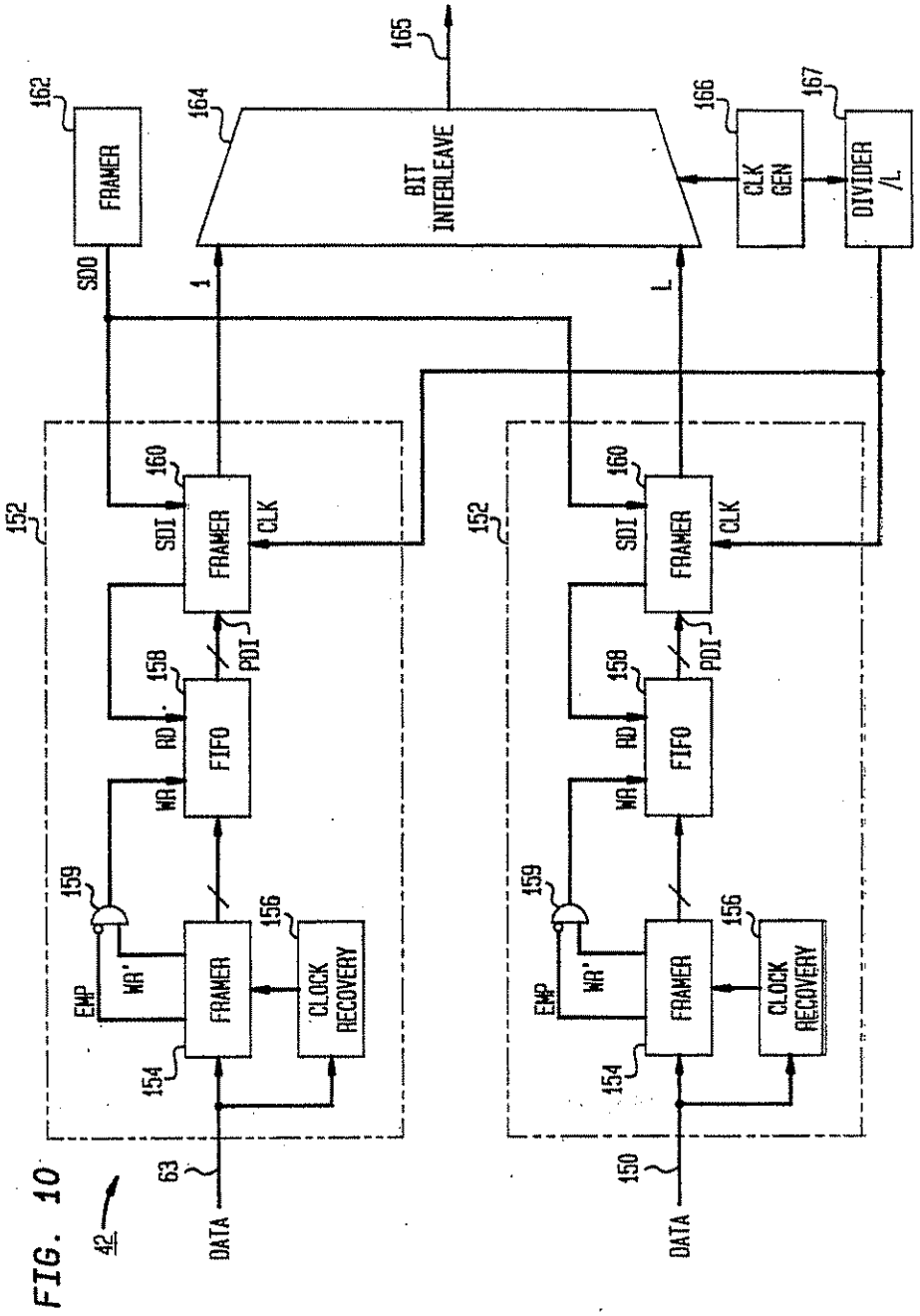

165
162
FRAMER
164
BIT INTERLEAVE
166
CLK GEN
167
DIVIDER /L
SDO
1
L
152
160
SDI
FRAMER
CLK
158
RD
FIFO
WR
PDI
159
EMP
WR'
156
CLOCK RECOVERY
154
63
DATA
150
42

FIG. 10

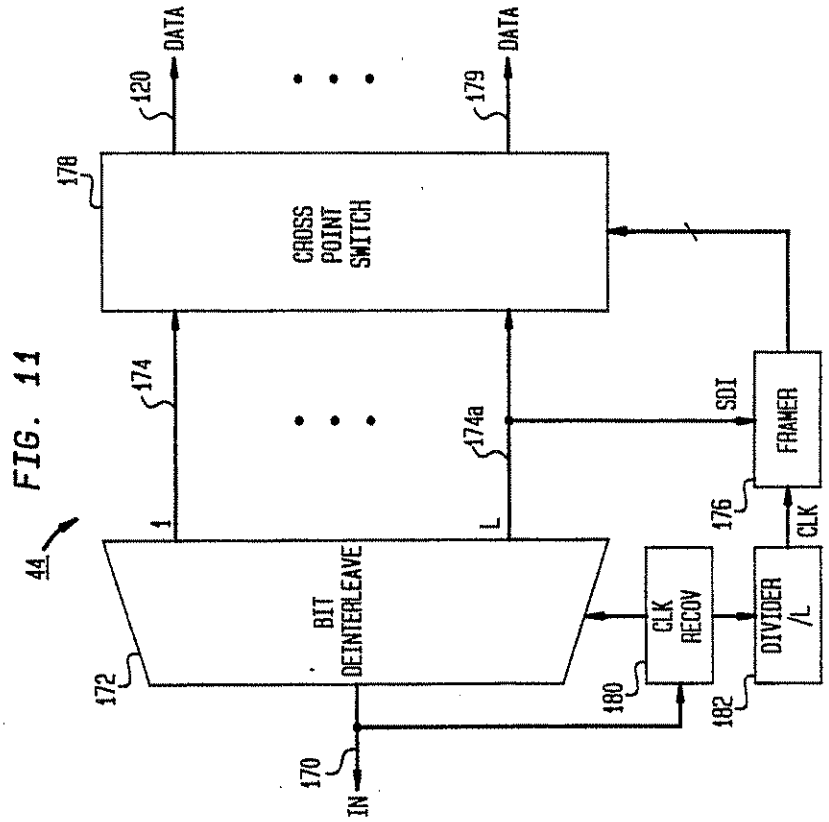

DATA
120
DATA
179
178
CROSS POINT SWITCH
174
174a
SDI
FRAMER
176
CLK
1
L
44
BIT DEINTERLEAVE
CLK RECOV
DIVIDER /L
172
180
182
170
IN

FIG. 11

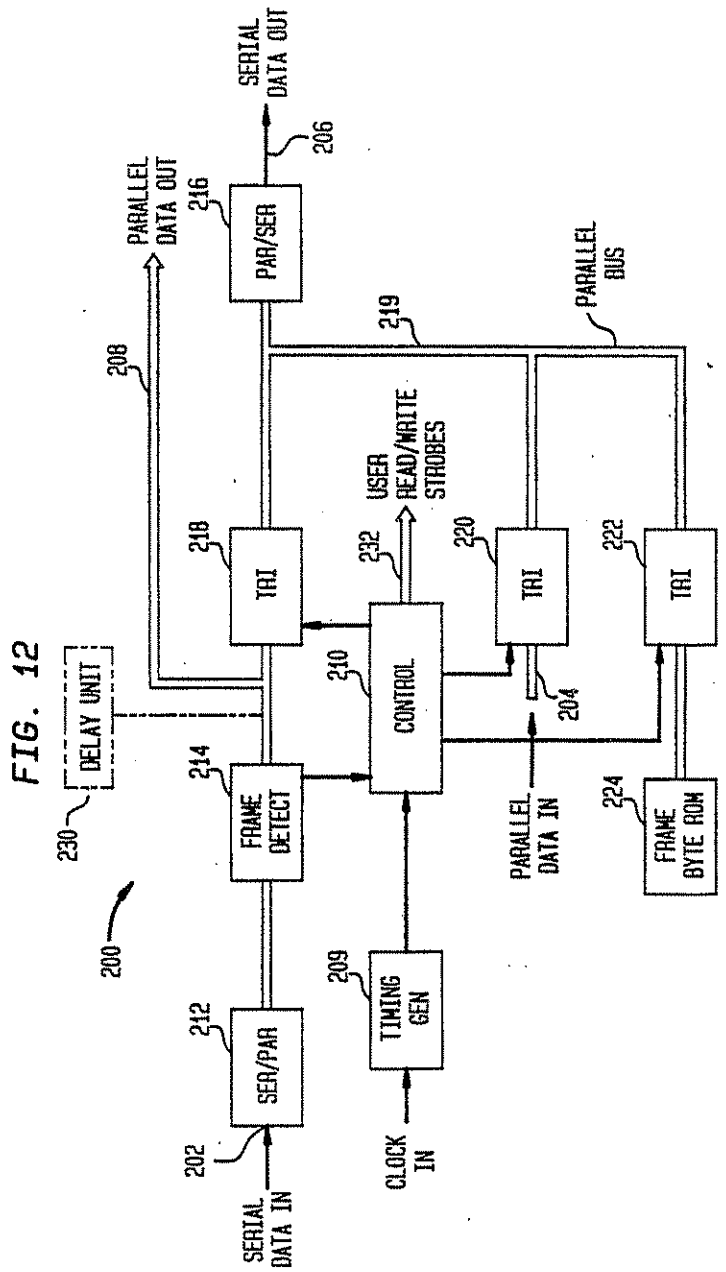

SERIAL DATA OUT
206
PARALLEL DATA OUT
216
PAR/SER
219
PARALLEL BUS
208
USER READ/WRITE STROBES
232
218
TRI
220
TRI
222
TRI
210
CONTROL
204
DELAY UNIT
230
214
FRAME DETECT
224
FRAME BYTE ROM
PARALLEL DATA IN
200
209
TIMING GEN
212
SER/PAR
202
CLOCK IN
SERIAL DATA IN

FIG. 12

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 constructing 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

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 DS3 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,

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 73*a*) 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-

mitted 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

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 10^5.

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 (SP ID) is inserted for each tributary before they are multiplexed at the transmit side. The tributary present on line 174*a* 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 1424–1431 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 or 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

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 **224** 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.

* * * * *

Exhibit B

Westlaw.

62 Fed.Appx. 951
62 Fed.Appx. 951
(Cite as: 62 Fed.Appx. 951)

H

Briefs and Other Related Documents

This case was not selected for publication in the Federal Reporter.

NOTE: Pursuant to Fed.Cir.R. 47.6, this order is not citable as precedent. It is public record.

Please use FIND to look at the applicable circuit court rule before citing this opinion. Federal Circuit Rule 47.6. (FIND CTAF Rule 47.6.)

United States Court of Appeals,
Federal Circuit.
BELL COMMUNICATIONS RESEARCH, INC. (now known as Telcordia Technologies, Inc.), Plaintiff-Appellant,
v.
FORE SYSTEMS, INC. (now known as Marconi Communications, Inc.), Defendant-Cross Appellant.
Nos. 02-1083, 02-1084.

March 27, 2003.

Owner of patents for multiplexing and demultiplexing digital data streams sued competitor for infringement. Competitor counterclaimed, asserting noninfringement, invalidity, and unenforceability. The United States District Court for the District of Delaware, Farnan, J., 113 F.Supp.2d 635, construed claims, and entered judgment of noninfringement. Owner appealed, and competitor cross-appealed. The Court of Appeals, Clevenger, Circuit Judge, held that: (1) claim for multiplexing patent did not require that generation of frames whose payload fields had no data in them be completed before filling of such fields with data could commence; (2) competitor could not bring conditional cross-appeal; and (3) claim for demultiplexing patent required "bit stream" in question to have been formed from two or more pre-existing base-level frames.

Affirmed in part, dismissed in part, vacated in part, and remanded.

Mayer, Chief Judge, dissented and filed opinion.

West Headnotes

[1] **Patents** 101(4)
291k101(4) Most Cited Cases

[1] **Patents** 168(2.1)
291k168(2.1) Most Cited Cases
"Filling" of "empty payload fields," as called for in patent for telecommunications device, did not require that generation of frames whose payload fields had no data in them be completed before filling of such fields with data could commence, even if at least part of frame had to be generated before it could begin receiving data, where specification disclosed features of overhead field that would permit filling to begin before generation was complete, and neither precedent, grammar, logic, specification, nor prosecution history dictated that complete frame had to be generated before filling process could begin.

[2] **Patents** 101(2)
291k101(2) Most Cited Cases
Telecommunications device patent, calling for creation of information packets "for transmitting data ... from ... plurality of sources simultaneously," did not require that data sources insert data into empty frames at same moment in time; claim spoke not of simultaneous insertion of data into empty payload fields, but of simultaneous transmission of data from several sources in bit stream.

[3] **Patents** 324.52
291k324.52 Most Cited Cases
Defendant who prevailed by obtaining judgment of noninfringement had no right to bring conditional cross-appeal to challenge claim construction rulings.

[4] **Patents** 165(4)
291k165(4) Most Cited Cases
Claim in telecommunications device patent reciting method operating on "interleaved multiplicity of data bytes ... from plurality of identically-formatted contributory frames" required "bit stream" in question to have been formed from two or more pre-existing base-level frames, where preamble stated that bit stream was "derived from a plurality of identically-formatted contributory frames."

Patents 328(2)

© 2006 Thomson/West. No Claim to Orig. U.S. Govt. Works.

291k328(2) Most Cited Cases
4,835,768, 4,893,306. Construed.
***952** Before MAYER, Chief Judge, CLEVENGER and BRYSON, Circuit Judges.

Opinion

CLEVENGER, Circuit Judge.

Bell Communications Research, Inc. ("Bellcore," now known as Telcordia Technologies, Inc.), appeals the judgment of the United States District Court for the District of Delaware, granting judgment of noninfringement of United States Patents No. 4,893,306 (" '306 patent") and No. 4,835,768 (" '768 patent") to FORE Systems, Inc. ("FORE," now known as Marconi Communications, Inc.). *Bell Communications Research, Inc. v. FORE Sys., Inc.*, No. 98-586-JJF (D.Del. Sept.21, 2000), *amended*, No. 98-586-JJF (D.Del. Sept.21, 2001). FORE cross-appeals to contest two of the district court's claim construction rulings on the '306 patent. We *vacate* the judgment of noninfringement of the '306 patent, *affirm* the judgment of noninfringement of the '768 patent, *dismiss* the cross-appeal, and *remand* the case for further proceedings.

BACKGROUND

Both the '306 and '768 patents relate to multiplexed data transmission protocols. The '306 patent is concerned with a method of dynamic time division multiplexing (DTDM), in which a single transmission line is shared among several data sources by allocating discrete segments, or "frames," of the bit stream to each data source. Rather than pre-assign partitions of the bit stream to each data source, the system described by the patent allocates frames to each data source dynamically, depending on the priority of each data source and whether each data source has data available for transmission.

According to the disclosure of the '306 patent, the bit stream is formed by generating a train of DTDM frames, each consisting of a "transmission overhead field" that contains information about the frame and marks its boundary, and a "payload field," which is initially empty. Incoming source data streams are broken into discrete segments, or packets, each of which has a header identifying from which data source it originates. Specialized "framer" circuits in a DTDM assembler then insert the packets individually into available payload fields of the DTDM bit stream, with priority among the data sources determined automatically by the proximity of each data source's framer to the origin of the empty bit stream.

The '768 patent is addressed to a slightly different form of multiplexing, employed with a signal hierarchy termed SONET (Synchronous Optical Network). SONET transmissions are structured around an 810-byte frame, which the parties refer to ***953** as a "base-level" frame. The lowest rate of SONET communication, STS-1, transmits data at 8,000 base-level frames per second, for a serial bit transmission rate of 51.84 megabits per second. Higher rates of data transmission are achieved by interleaving multiple STS-1 frames together into a larger, "higher-level" frame. These higher-level frames are referred to generally as STS-N frames; a particular designation (such as STS-24) means a frame with that number of STS-1 frames interleaved.

The '768 specification discloses circuitry and algorithms to perform steps in preparation for separating STS-N signals into lower-level frames. In particular, the specification describes two preliminary functions that may be performed by high-speed integrated circuits. The first function is conversion of the incoming serial data stream to parallel (byte-wide) form, also referred to as synchronization of byte formatting. In order to format the serial bit stream into data bytes with the appropriate byte registration, the protocol described in the specification identifies a signature byte, F1, that appears once in each STS-1 frame. Identification of the F1 byte in the bit stream allows the circuitry to divide the incoming bit stream into bytes with the same boundaries as the bytes of the original transmission. The second function disclosed in the specification is the identification of a "benchmark" occurring once in each higher-level frame. In a SONET STS-N frame, this benchmark is the three-byte sequence F1F2F2, marking the transition between the interleaved F1 framing bytes and the interleaved F2 framing bytes of the STS-1 frames. The ' 768 specification teaches that the F1F2F2 benchmark may be used to determine the boundaries of frames in the bit stream, and discloses circuitry that monitors each frame for the F1F2F2 benchmark in order to ensure that byte synchronization is maintained.

Bellcore filed suit against FORE for infringement of the '306 and '768 patents, as well as two counts for infringement of other patents that have been dismissed. FORE counterclaimed, asserting noninfringement, invalidity, and unenforceability. The district court held a *Markman* hearing, and issued an opinion and order construing disputed

© 2006 Thomson/West. No Claim to Orig. U.S. Govt. Works.

claim terms, with a supplemental order on a means-plus-function claim of the '306 patent.

After the district court construed the claims, Bellcore advised the court that it could not prevail under the court's claim construction. Bellcore requested that the court either certify an interlocutory appeal under Rule 54(b), or enter judgment of noninfringement and dismiss FORE's counterclaims without prejudice as moot. Bellcore did not identify precisely which claim constructions precluded infringement. Over FORE's opposition, the district court complied with Bellcore's request, entering judgment of noninfringement in favor of FORE and dismissing FORE's counterclaims without prejudice as moot. The object of this exercise was to permit early review by this court of the claim constructions that precluded Bellcore from asserting infringement of the '306 and '768 patents.

FORE protested the form of the judgment, because Bellcore had not identified the relevant claim constructions that would be disputed on appeal. In response, Bellcore stipulated that the construction given to three limitations of the '306 claims and a portion of the '768 claim 13 preamble precluded a finding of infringement. The district court then granted FORE's motion to amend the judgment "to incorporate [Bellcore's] ... concessions."

Bellcore appeals the judgment of noninfringement entered against it, based on the ***954** district court's claim constructions identified in Bellcore's stipulations. Bellcore also appeals other claim constructions which do not appear to have been addressed in the stipulations. FORE cross-appeals to argue additional claim limitations of the '306 patent that were construed against it below, stating that its cross-appeal is conditional upon this court resolving the appeal on the '306 patent in Bellcore's favor. We exercise jurisdiction over the appeals pursuant to 28 U.S.C. § 1295(a)(1).

I

Claims 1, 3, and 4 of the '306 patent are at issue in this appeal. Because the district court construed the terms of claims 1, 3, and 4 in conformity with each other, all the issues disputed by the parties are common to all three claims. Essentially all of the disputed language appears in claim 1:

> 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.

As a question of law, we review the district court's claim construction without deference. *Cybor Corp. v. FAS Techs., Inc.*, 138 F.3d 1448, 1456, 46 USPQ2d 1169, 1174 (Fed.Cir.1998) (*en banc*). In light of Bellcore's stipulation that it cannot establish infringement without prevailing on its claim construction arguments, we confine our review to the district court's claim construction rulings.

A

The first and most salient dispute over construction of the '306 claims is whether a complete frame must be generated before a framer may begin filling it with data (FORE's position), or whether the framer can begin filling the "front" part of the frame with data while the "rear" end is still being generated (Bellcore's position). Although the district court's claim construction does not explicitly require that a "complete" frame must be generated before the payload fields are filled with data, we accept the parties' interpretation that the district court so held. Thus, the question before us is whether the first claim step ("generating") must be completed, for at least one frame, before the second step ("filling") can begin.

[1] We conclude that it does not. FORE, citing several cases in which we have construed method claims to require sequential performance of their steps, seems to suggest a general principle that method claims should be construed to require sequential performance of their recited steps. The precise question here is not whether the first step must be performed before the second step is performed, but whether it must be *completed* before the second step is begun. Regardless, as we recently reiterated in *Altiris, Inc. v. Symantec Corp.*, 318 F.3d 1363, 1369-71, 65 USPQ2d 1865, 1869-70 (Fed.Cir.2003), the steps of a method claim need ***955** not be performed in the order written unless

© 2006 Thomson/West. No Claim to Orig. U.S. Govt. Works.

logic, grammar, or the content of the specification dictates otherwise. *See also* Interactive Gift Express, Inc. v. Compuserve Inc., 256 F.3d 1323, 1342, 59 USPQ2d 1401, 1416 (Fed.Cir.2001) ("Unless the steps of a method actually recite an order, the steps are not ordinarily construed to require one.").

In the case of the '306 patent, neither logic, grammar, nor the specification compels the conclusion that "generating" must be complete before "filling" may begin. It is undisputed that some "generating" must precede "filling," because at least part of a frame must be generated before it can begin receiving data. But unlike railroad boxcars, in which filling of a partially generated car might pose some problems, frames are capable of receiving data even if they are only partially generated. Consequently, logic does not demand that the filling process must wait until the rear boundaries of the frame have been generated.

Nor does grammar demand such a result. Because both "generating" and "filling" are continuous and concurrent processes in the method of the claims, it makes little sense to speak of the generating process being "complete" before filling begins. Indeed, as Bellcore notes, a strict grammatical requirement that the first step be complete before the second begins would lead to an unreasonable interpretation of the claim. The first step of the claim recites generation not of a single frame, but a "bit stream comprising a sequence of frames." If the first step must be completed before the second step begins, then all the frames, not just one, must be generated before any data may be inserted into the bit stream. While such a mode is theoretically possible, it would be manifestly unsuitable for continuous transmission of data in a telecommunications network.

Nor does the specification teach that generating must be completed before filling can begin. It is fair to say that the specification is silent on the subject of whether frames are generated "byte-by-byte," as Bellcore suggests, or whether complete frames are generated before data insertion begins, as FORE contends. FORE argues that Figures 2 and 4 of the specification indicate that empty frames must be generated before they can be filled. However, we agree with Bellcore that Figure 2 depicts the claimed process only schematically, not literally. While Figure 2 does show a train of pre-generated frames entering a "DTDM assembler," Figure 2 obviously does not attempt to explain how frames arise, because Figure 2 shows frames arriving from an undepicted source. Moreover, despite FORE's argument to the contrary, Figure 4 of the patent clearly depicts a "DTDM assembler" comprising a framer (52) that generates empty frames. Thus, to interpret Figure 2 literally would set it at odds with Figure 4: Figure 4 indicates that a "DTDM assembler" generates empty frames internally, instead of receiving them from an external source as depicted in Figure 2. Figures 2 and 4 therefore do not compel FORE's interpretation.

Nor does the specification describe any circuitry or algorithm that would delay the filling process until a complete empty frame was generated or received, or any indication that frames are passed from framer to framer as frame-long chunks of data rather than byte-by-byte. Such features might be expected if FORE's interpretation were correct. However, the specification does disclose a feature that would allow a framer to begin filling frames before generation was complete. According to the specification, each frame ***956** comprises a "transmission overhead field" containing information about the contents of the frame. Included in the overhead field may be a flag indicating whether the frame is empty or full. '306 patent, col. 6, ll. 61- 65. Because the transmission overhead field is shown preceding the payload field in the bit stream, *see id.* Fig. 1, a framer can determine whether an incoming frame is "empty" or "full" when it receives the overhead field, even if the remainder of the frame has yet to be generated. Accordingly, the embodiment described in the specification permits a framer to begin inserting data once it has received the overhead field and the first bytes of the payload field, without complete generation of an empty frame. This description supports Bellcore's interpretation.

Finally, FORE contends that Bellcore limited itself to a "sequential" interpretation of the claims during prosecution. When distinguishing the pending claims from the prior art (the Baran reference), Bellcore described the claimed method as "first generating a bit stream comprised of frames," and "then" inserting packets "into the empty payload fields of the frames." Statements in the prosecution history will limit claim terms to exclude interpretations disclaimed during prosecution. Southwall Techs., Inc. v. Cardinal IG Co., 54 F.3d 1570, 1576, 34 USPQ2d 1673, 1676-77 (Fed.Cir.1995). However, no such disclaimer took place here. The Baran reference which Bellcore was traversing used a completely different system of multiplexing. Bellcore did not disclaim coverage to concurrent generating and filling to overcome Baran, nor did Bellcore distinguish Baran by arguing that its invention required complete generation of the frames before filling could begin. The prosecution history therefore does not address whether generation of an

© 2006 Thomson/West. No Claim to Orig. U.S. Govt. Works.

entire frame must be completed before filling begins.

In summary, neither precedent, grammar, logic, specification, nor prosecution history dictates that a complete frame be generated before the filling process may begin. Given that the specification discloses features of the overhead field that would permit filling to begin before generation is complete, it would be error to impose upon the claims a requirement that filling cannot begin until one or more empty frames are generated completely. We therefore agree with Bellcore that the claims encompass the insertion of data into a frame's empty payload field while the frame is still being generated.

B

The second dispute over construction of the asserted claims of the '306 patent concerns the requirement imposed by the district court that "two or more empty frames are filled at the same time by different data sources." The district court drew this requirement from the preamble language, "for transmitting data from said plurality of sources simultaneously via said bit stream." The dispute, as presented by the parties, is whether this limitation requires two or more data sources to be inserting data into empty frames at the same moment in time.

[2] We agree with Bellcore that data sources need not insert data into empty frames at the same moment in time. The claim speaks not of simultaneous insertion of data into empty payload fields, but of the simultaneous transmission of data from several sources in the bit stream. Perhaps the term "simultaneously" is ambiguous because it could refer either to events taking place at the same moment in time, or to events that both take place within a defined interval of time. But reference to the specification unquestionably ***957** shows that the latter meaning is correct. The entire '306 patent is directed to time division multiplexing, and the essence of time division multiplexing is that a single communications line transmits the signals from two or more sources by allocating sequential portions of the bit stream to the competing input sources. '306 patent, cols. 1-2, 5. While any given point of the bit stream is dedicated exclusively to a single input source, over an interval of time (determined by the bit rate of the slowest data source) the bit stream will carry data from all the input sources. Hence, a time division multiplexed signal carries several input signals "simultaneously" without regard to the precise timing of data insertion.

The district court drew the opposite conclusion-interpreting "simultaneously" to mean at the same moment in time-from the specification's single use of the term "simultaneously" to describe the operation of demultiplexing circuitry. '306 patent, col. 13, ll. 49-51. But this portion of the specification refers to an entirely different aspect of the invention. Moreover, as Bellcore notes, the disclosure nowhere suggests that data is inserted into two or more empty frames at the same moment in time. According to the specification, the timing with which interface units insert data into the bit stream is controlled solely by whether the interface unit has data stored in its FIFO, and whether upstream interface units have left empty frames available in the bit stream. *Id.* col. 9, ll. 50-53; col. 7, ll. 46-61. There is no mention of any mechanism to coordinate simultaneous insertion of data.

FORE points out that it is *possible* for multiple data sources to insert data simultaneously into the bit stream, if two or more interface units have data stored in their FIFOs and have empty frames positioned in their framer units at the same time. But of course, a claimed invention is not limited to a particular mode of operation simply because it is capable of operating in that mode. We conclude that "simultaneously" in the claim preamble refers to the capability of the multiplexed bit stream to carry signals from multiple sources during a finite interval, not to any requirement that several data sources must be inserting data into empty frames at the same instant in time.

C

The third issue raised by Bellcore illustrates the difficulties posed by what are essentially interlocutory appeals from district courts' claim construction orders. The district court construed "empty payload field" to mean that "a frame's payload has zero data in it," rejecting Bellcore's argument that "empty payload field" means any condition representing an absence of source data. Bellcore is not certain exactly what the district court meant by "zero data." Nonetheless, Bellcore has appealed the district court's construction of this term, stating that it cannot prevail if the district court actually meant "no bit signals of any kind" when it said "zero data."

We decline to play Prophetes to the district court's Pythia. At oral argument, FORE stated its understanding that "zero data" encompasses various bit signals that might maintain the stated transmission rate of a bit stream, including "placeholders" or "garbage bits." Because Bellcore conceded

© 2006 Thomson/West. No Claim to Orig. U.S. Govt. Works.

noninfringement only under a definition of "zero data" restricted to "no bit signals of any kind," the parties' agreement on a broader meaning for "zero data" removes this limitation as grounds for noninfringement, at least at this stage of the proceedings. We need not speculate further on the district court's meaning, nor ***958** refine its construction of this limitation any further on appeal.

D

[3] FORE's cross-appeal addresses two additional limitations appearing in the '306 claims, "bit stream" and "frame timing information." We dismiss the cross-appeal, *sua sponte,* as improper. The district court entered judgment of noninfringement in favor of FORE and dismissed FORE's counterclaims as moot. FORE has not objected to this procedure on appeal. A prevailing party has no right of cross-appeal, *Lindheimer v. Ill. Bell Tel. Co.*, 292 U.S. 151, 176, 54 S.Ct. 658, 78 L.Ed. 1182 (1934), and a defendant who prevails on noninfringement has no right to bring a "conditional" cross-appeal to challenge claim construction rulings. *Bailey v. Dart Container Corp. of Mich.*, 292 F.3d 1360, 63 USPQ2d 1319 (Fed.Cir.2002).

Yet FORE, despite its prevailing party status, is entitled to argue those claim constructions on which Bellcore prevailed as alternative grounds for affirming the district court's judgment. *United States v. Am. Ry. Express Co.*, 265 U.S. 425, 435, 44 S.Ct. 560, 68 L.Ed. 1087 (1924); *Bailey*, 292 F.3d at 1362, 63 USPQ2d at 1320. However, we cannot determine from FORE's submissions whether the district court's judgment of noninfringement could be upheld if we were to agree with FORE's additional claim construction arguments. Bellcore has not conceded noninfringement under the two claim constructions advocated in FORE's cross-appeal, and FORE makes no attempt to demonstrate that a ruling in its favor would preclude Bellcore from establishing infringement of the '306 patent. FORE's argument on "frame timing information," in which FORE argues for a broad construction while Bellcore argues for a narrow one, may even be directed to one of FORE's dismissed invalidity counterclaims. We will not resolve disputes which we cannot relate to a case or controversy under the patent laws, and we therefore decline to address the claim construction arguments made in FORE's cross-appeal. FORE is not precluded from disputing the district court's construction of these limitations in the future.

II

Claim 13 of the '768 patent is the only claim asserted by Bellcore against FORE. We think that both parties have misconstrued claim 13. Claim 13 recites a method operating on

> 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.

The parties assert in their briefs that claim 13's "contributory frames" are "base-level" frames that may be interleaved into a higher-level frame, *i.e.* STS-1 frames that may be interleaved into an STS-N frame. However, the claim speaks of only one kind of frame, and those frames might well be "contributory" to the recited *bit stream* instead of "contributory" to unrecited higher-level frames. Further, step (e) of claim 13 refers to identifying a "contiguous plurality of bytes" present in "each of said contributory frames." In the context of the SONET benchmark identification process disclosed by the specification, the "contiguous plurality of bytes" is the sequence F1F2F2. This sequence is present in each STS-N frame, but not in an STS-1 frame. '768 patent, col. 5, ll. 47-62. Indeed, the entire utility of using the F1F2F2 signal as a benchmark rests on the principle that the F1F2F2 sequence repeats once each STS-N frame. *Id.* col. ***959** 5, ll. 53-57. This disclosure cannot be reconciled with the parties' interpretation of step (e), which equates "contributory frame" with "base-level frame." We think this inconsistency reflects not a "glitch" or "mistake" in the claim (as suggested by the parties), but an error in the premise that "contributory frame" means "base-level frame."

Nonetheless, at oral argument, both parties were steadfast in their insistence that "contributory frames" in claim 13 means base-level frames such as STS-1, not higher-level frames such as STS-N. Accordingly, rather than impose our own interpretation of "contributory frame" upon the case, we will decide the claim construction dispute on the grounds set forth by the parties, and under their definition of "contributory frame" as a base-level frame.

Bellcore advances, again with some uncertainty, the view that the district court limited claim 13 to require "pre-existing contributory frames." In terms of the SONET protocol, this translates to a requirement that STS-1 frames exist as distinct entities prior to being interleaved into an STS-N frame. On the premise that the district court so limited the claim, Bellcore argues that the district court erred.

[4] We affirm the district court's claim construction.

© 2006 Thomson/West. No Claim to Orig. U.S. Govt. Works.

The district court's claim construction order makes clear that it did require the "bit stream" in question to have been formed from pre-existing base-level frames. The district court stated that the preamble of claim 13:

> means 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 serial bit stream that is being demultiplexed must have been formed by interleaving the bytes of two or more contributory frames.

This language clearly indicates that the base-level frames must have had a separate existence prior to being interleaved into a higher-level frame.

Regardless of whether the district court was correct to require the actual separation of the higher-level frame into its base-level constituents, or whether the district court limited claim 13 to the SONET protocol, we hold that under the parties' definition of "contributory frame" the district court correctly required the bit stream to have been formed from pre-existing contributory frames.

We agree with FORE that this conclusion must follow from the preamble phrase "derived from a plurality of identically-formatted contributory frames." "Derive" is best defined here as "to have or take origin: ORIGINATE: STEM, EMANATE." *Webster's Third New International Dictionary* 608 (1993). A thing cannot originate from a source that has never existed. Consequently, a bit stream cannot be derived from base-level frames unless those frames first existed. Bellcore presents from the same dictionary the competing definition "to trace the origin, descent or derivation of," and argues that this definition supports a broader reading of "derived." However, Bellcore has put forth a definition of "derive" in its transitive sense rather than its intransitive one, and simple grammar precludes Bellcore's interpretation. In claim 13's preamble, the subject of "derived" is "bit stream," or perhaps "data bytes," and "derived" as a transitive verb would require a direct object not to be found in the claim language. Put more simply, the bit stream or data bytes of the preamble are not busy "tracing" their origin or descent, or being so traced, from anything. Claim 13 refers to a method of multiplexing, not a method of genealogy.

***960** Having concluded that the plain meaning of "derived from" supports FORE's position, we find no indication that the term actually means "following the format of," as Bellcore contends. The specification is not particularly concerned with how the bit stream was generated, although as FORE and the district court noted, the descriptions of multiplexing and demultiplexing in the specification do refer to assembly of the bit stream from STS-1 frames, and its disassembly into STS-1 frames. '768 patent, col. 1, ll. 37-44; col. 4, ll. 9-13; col. 5, ll. 45-47. But while the algorithms disclosed in the specification might function properly on a bit stream formatted as if it were assembled from STS-1 frames, we find no indication in the specification that the claim language "derived from a plurality of identically-formatted contributory frames" has any meaning other than its plain meaning. The district court therefore correctly construed this claim language to require that the bit stream be assembled from two or more pre-existing base-level frames.

Bellcore's stipulation of noninfringement, or at least that portion of it which the parties have provided to us, does not speak directly of a requirement for pre-existing base-level frames. However, Bellcore concedes that it must prevail on all its claim construction arguments in order to prevail on each patent, and Bellcore has unequivocally framed the dispute in terms of whether claim 13 requires the bit stream to be generated from pre-existing base-level frames. Moreover, from what meager information we have regarding the accused device, it appears that FORE's system does not build up a bit stream from pre-existing base-level frames. The district court's entry of judgment of noninfringement of claim 13 of the '768 patent may therefore be sustained on the basis of this holding alone. In light of this disposition, we need not address the other claim construction issues raised by Bellcore in its appeal.

CONCLUSION

For the reasons set forth above, we affirm the district court's judgment of noninfringement of the '768 patent, but vacate the district court's judgment of noninfringement of the '306 patent and remand the case for further proceedings. We dismiss FORE's cross-appeal as improper.

COSTS

No costs.

MAYER, Chief Judge, dissenting.

While I do not take issue with the court's construction of the simultaneous transmission language or the empty payload field requirement, Bell Communications ("Belcore") conceded that under the district court's claim construction, it could

© 2006 Thomson/West. No Claim to Orig. U.S. Govt. Works.

not prove infringement of these limitations as well as the frame limitation of claims 1, 3, and 4 of United States Patent No. 4,893,306 (" '306 patent"). Because the district court correctly concluded that the '306 patent requires generation of a complete frame prior to filling, however, I would *affirm* the judgment of non-infringement.

Claim 1 of the '306 patent requires, inter alia, *"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 ..." '306 patent, col. 17, ll. 47-53, (emphasis added). While the steps of method claims may not necessarily have to be performed sequentially or each step fully completed before the next begins, there are method claims that require full ***961** completion of each step prior to the start of the next and a strict adherence to the sequential order. The plain meaning of the claims themselves, the place that all claim construction must begin, mandates that the generating step must be completed prior to the beginning of the filling step. *CCS Fitness, Inc. v. Brunswick Corp.*, 288 F.3d 1359, 1366, 62 USPQ2d 1658, 1662 (Fed.Cir.2002) (There is a heavy presumption that the plain meaning of the claim language controls and can only be overcome in limited circumstances.).

"Fill" is defined as "to supply with as much as can be held or contained; to put or pour something into (a receptacle) till no more can be received." 5 Oxford English Dictionary 908 (2nd ed.1989). The Webster's Dictionary provides almost the identical definition. Webster's Third New International Dictionary 849 (1993). Logic dictates that an incomplete container or frame cannot be filled because filling requires a finite volume. The patentee could have chosen alternative language to convey partial frame filling, whether by expressly stating the possibility or using a verb such as "placing" which would not require a finite volume, but did not do so. The prosecution history confirms this construction. In overcoming the Baran reference, the patentee stated that the claims require *"first generating* a bit stream comprised of *frames* with empty payload fields.... The packets are *then inserted* into the empty payload fields ..." The patentee stated that first frames, plural, are generated and then data is inserted. The patentee makes no mention of generating the transmission overhead field of a frame and the first boundary of the payload field and then inserting data on a byte-by-byte basis. And the fact that Baran uses a different type of multiplexing is of no matter. Clear assertions made during prosecution in support of patentability, whether or not actually required to secure allowance of the claim, can create an estoppel. *Texas Instruments, Inc. v. United States Int'l Trade Comm'n*, 988 F.2d 1165, 1174, 26 USPQ2d 1018, 1025 (Fed.Cir.1993).

The court frames the issue as whether the specification or the prosecution history obviates the construction that filling could begin into a partially generated frame and answers in the negative. While the specification may not expressly exclude such a construction, nothing supports it either. The specification only speaks of generated frames and is silent as to anything less.

The court cites to a paragraph in the specification and argues that it presents a possible mechanism for byte-by-byte insertion prior to completion of the frame. *Ante,* at 955-956. The cited portion of the specification states: "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." '306 patent, col. 6, ll. 61- 65. This statement offers nothing more than the components that make up a frame and certainly does not provide support for partial frame filling. While the court is correct that it is fair to say that the specification is silent about whether frames are generated byte-by-byte or whether complete frames are generated before data insertion begins, the patentee should bear the burden for the lack of information and the plain meaning of the language chosen by the patentee should control.

62 Fed.Appx. 951

Briefs and Other Related Documents **(Back to top)**

• 2002 WL 32625311 (Appellate Brief) Reply Brief for Defendant-Cross Appellant Fore Systems, Inc. (Aug. 09, 2002)Original Image of this Document (PDF)

• 2002 WL 32625312 (Appellate Brief) Reply Brief for Plaintiff-Appellant Bell Communications Research, Inc. (Jul. 23, 2002)Original Image of this Document (PDF)

• 2002 WL 32712437 (Appellate Brief) Brief for Defendant-Cross Appellant Fore Systems, Inc. (May.

© 2006 Thomson/West. No Claim to Orig. U.S. Govt. Works.

28, 2002)Original Image of this Document (PDF)

• 2002 WL 32625313 (Appellate Brief) Brief for Plaintiff-Appellant Bell Communications Research, Inc. (Mar. 18, 2002)Original Image of this Document with Appendix (PDF)

• 02-1084 (Docket) (Dec. 14, 2001)

• 02-1083 (Docket) (Dec. 14, 2001)

END OF DOCUMENT

© 2006 Thomson/West. No Claim to Orig. U.S. Govt. Works.

Exhibit C

02-1083, -1084

PUBLIC COPY

IN THE

UNITED STATES COURT OF APPEALS

FOR THE FEDERAL CIRCUIT

FILED
U.S. COURT OF APPEALS FOR
THE FEDERAL CIRCUIT
MAR 18 2002
JAN HORBALY
CLERK

BELL COMMUNICATIONS RESEARCH, INC.
(now known as Telcordia Technologies, Inc.),

Plaintiff-Appellant,

v.

FORE SYSTEMS, INC.
(now known as Marconi Communications, Inc.),

Defendant-Cross Appellant.

Appeals from the U.S. District Court for the District of Delaware in 98-CV-586, Judge Joseph J. Farnan, Jr.

BRIEF FOR PLAINTIFF-APPELLANT BELL COMMUNICATIONS RESEARCH, INC.

Of Counsel:

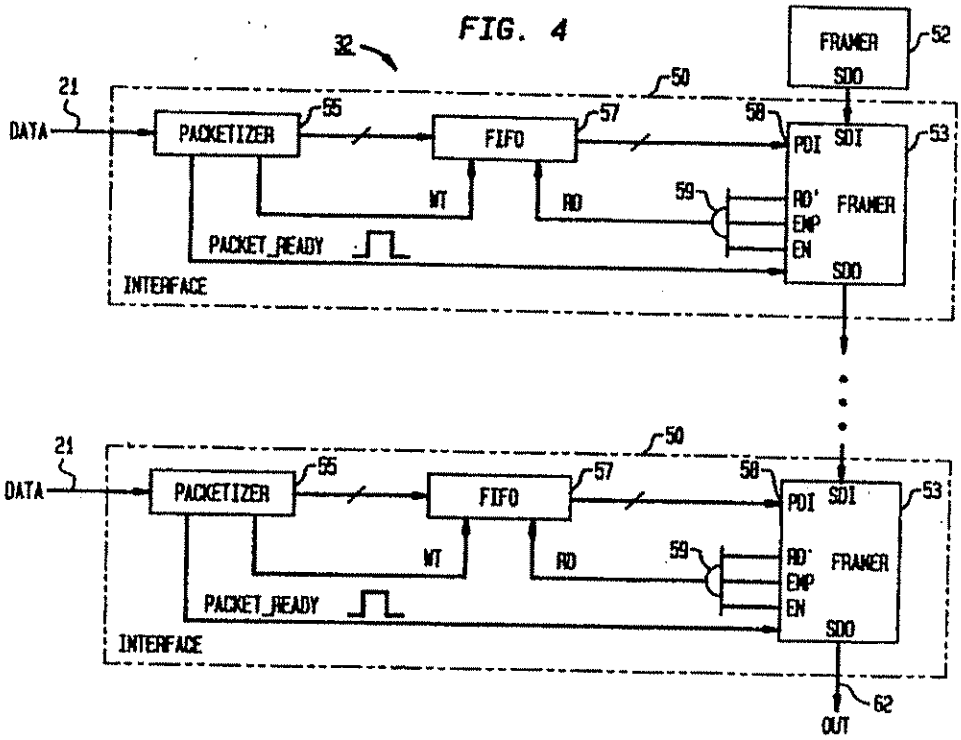
Marcelle E. Mihaila
Richard W. McAlister
GRAY CARY WARE & FRIEDENRICH, LLP
999 Third Avenue, Suite 4000
Seattle, Washington 98104-4033
(206) 839-4800

March 18, 2002

Donald R. Dunner
Don O. Burley
Richard H. Smith
Vincent P. Kovalick
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER, L.L.P.
1300 I Street, N.W., Suite 700
Washington, D.C. 20005-3315
(202) 408-4000

Attorneys for Plaintiff-Appellant Bell Communications Research, Inc.

FIG. 4



Frames are generated continuously, but on a byte-by-byte basis rather than as fully formed, complete units. Generating framer 52 generates the bytes that make up the frames. The multiple inserting framers 53 are linked in a "daisy chain" so that, as the bytes of each frame are generated by framer 52, they flow in the bit stream to the first framer 53, the second framer 53, and so on. A "packet" of data is accumulated from a different tributary source in each FIFO 57.[2] When a packet is

[2] Source data is transmitted as "packets," which are fixed numbers of data bytes to which control (e.g., routing) information has been attached. (A142, col.4/ll.54-60; A145, col.9/ll.18-25.) The FIFO (first-in, first-out) registers receive

(footnote continued on next page)

various sources "on a first-come, first-served basis." (*Id.*, col.5/ll.22-26.) Each source occupies only the number of empty frames needed to maintain its own bit rate. (*Id.*, col.5/ll.26-37.) Thus, if multiple inserting framers (the framers 53 in Figure 4) have packets of data ready to be transmitted at the same time, the specification teaches that the first of these framers will insert its data packet into the frame's empty payload field and change the frame's occupancy indicator to "full." The other inserting framers will then have to wait for another empty frame to insert their data. The specification calls this process, which makes simultaneous data insertion impossible, "the contention for empty DTDM frames." (A145, col.9/ll.50-53.)

The prosecution history supports this understanding of the specification. After changing language in claim 1 from "[a] method for transmitting circuit and packet data in a telecommunications network" to "[a] method for *simultaneously* transmitting data *from sources having different bit rates* in a telecommunications network" (A1063 (emphasis added)), Bellcore told the Patent & Trademark Office ("PTO") that the purpose of the patent involved the simultaneous transmission of data from diverse communication sources. Using the example from the specification in which the transmission stream is shared by three sources—each having a different bit rate and thus occupying a different number of frames—Bellcore explained that, "*In this manner*, data from multiple diverse sources