(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
31 January 2002 (31.01.2002)

PCT

(10) International Publication Number
WO 02/09351 A2

(51) International Patent Classification[7]: H04L 12/00

(21) International Application Number: PCT/US01/20748

(22) International Filing Date: 29 June 2001 (29.06.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/220,940 26 July 2000 (26.07.2000) US

(71) Applicant *(for all designated States except US)*: **THOMSON LICENSING S.A.** [FR/FR]; 46, quai Alphonse Le Gallo, F-92648 Boulogne Cedex (FR).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **LOCKRIDGE, Terry, Wayne** [US/US]; 10350 Ruckle Street, Indianapolis, IN 46280 (US). **HORLANDER, Thomas, Edward** [US/US]; 6234 Haverford Avenue, Indianapolis, IN 46220 (US). **JONES, Thomas, Herbert** [US/US]; 442 East Wood, Westfield, IN 46074 (US).

(74) Agents: **TRIPOLI, Joseph, S.** et al.; Thomson multimedia Licensing Inc., P.O. Box 5312, Princeton, NJ 08540 (US).

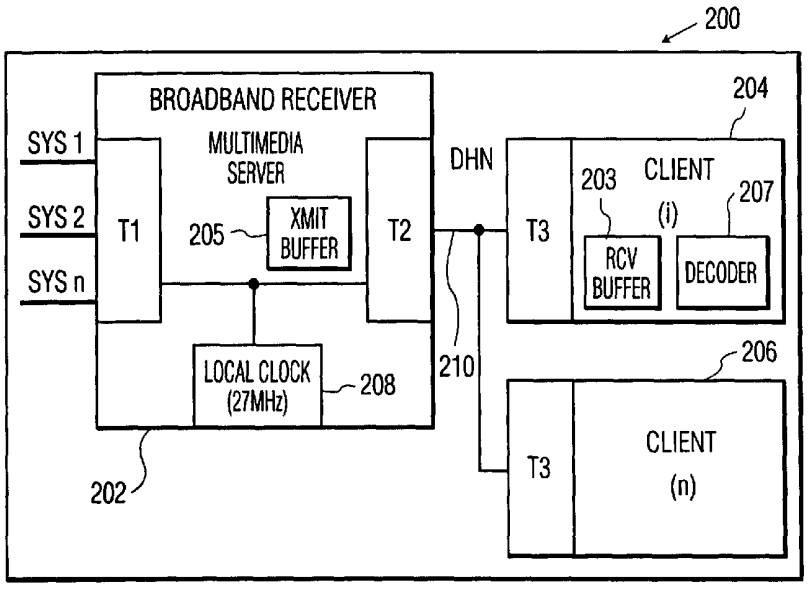
(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

[Continued on next page]

(54) Title: MULTI-MEDIA JITTER REMOVAL IN AN ASYNCHRONOUS DIGITAL HOME NETWORK



(57) Abstract: A method and an apparatus using a system level clocking scheme to remove jitter from multi-media packets distributed over an asynchronous network. The present invention overcomes the problems associated with jitter introduced in an asynchronous network by using various time stamps to synchronized a client device clock to a headend clock and to control the data flow in the client device to match the rate that the data is received by a broadband receiver coupled to the headend. The present invention allows the client device to synchronize to a selected one of a plurality of headend clock by including a clock adjustment factor along with the time stamps. The time stamps are added at the physical layer so that the time stamps correspond to the time the data packets are placed onto and received from the asynchronous network.

WO 02/09351 A2

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Multi-Media Jitter Removal in an Asynchronous Digital Home Network

The present invention relates to an asynchronous communications network, and in particular, to an asynchronous multi-media digital home network having a system level clocking scheme to remove jitter from multi-media data packets.

Recent advances in digital communications technology have increased the viability of digital home networks ("DHN"), which can allow various devices in a home to communicate with each other. In particular, the growing availability of broadband access has made it desirable to be able to tie in various devices in a home to a single gateway device that is coupled to a broadband network, or a plurality of networks, for centralized access and distribution of multi-media information. In such a digital home network, the gateway device accesses and buffers multi-media content and distributes the content over the network as requested by various client devices coupled to the network.

Ethernet is a common option for implementing a LAN. Ethernet is an asynchronous collision detect network. HPNA is also an asynchronous collision detect network and some power line based networks are often asynchronous collision detect networks. However, difficulties may arise when such networks are used to distribute multi-media content.

Fig. 1 is a system diagram illustrating a multimedia network. System 100 comprises a plurality of devices 108-113 coupled to media server 103 via one of a plurality of communications networks 105-107. In system 100, media server 103 receives multi-media content via satellite dish 102, buffers the content, and distributes the content as requested from the various devices coupled to the network. Media server 103 may also be coupled to receive multi-media content via any one of plurality of known digital media access methods, including, cable, and DSL. The transfer of data over the selected network is generally achieved by using large buffers or by allowing a client device to throttle data.

In this regard, there are a number of problems associated with taking a live broadcast and distributing it on an asynchronous digital home network. In a live broadcast signal, the data is pushed to the client device, and the client device does not have any control over the incoming data stream. In order to ensure that the

networked client device can decode and display all of the audio and video data delivered to it at the correct frame rate, and without repeated or dropped frames, it is in general required that the networked client device be clock synchronized to the headend clock. The most widely accepted method for clock synchronization in a broadcast A/V network relies on delivering counter samples taken at the broadcast site to the receiver with a fixed delay from the time of broadcast till the time of receipt. The counter at the broadcast site is incremented by the broadcast site's reference clock. At the instant those clock samples arrive in the receiver, a counter clocked by a voltage controlled oscillator (VCXO) is sampled. The value of the counter and the time stamp received from the broadcast site are compared and if the difference between the samples of the local counter and the clock samples taken at the broadcast site varies over time, the voltage applied to the local VCXO is adjusted to try to frequency lock the local clock to the broadcast site's reference clock.

On an asynchronous collision detect network, such as an Ethernet network, the delay from the time a packet is constructed and placed in a transmit buffer until the time the packet is received by a network connected device is not always constant. In such networks, multiple devices colliding on a network can cause variable packet and unpredictable delays in the reception of a packet. In this environment it is not possible to use the method described above to frequency lock the clock at the networked receiver to a clock at the networked transmitter due to the variable and unpredictable delays. Using such a method to attempt to lock the receiver clock to the transmitter clock may produce significant jitter in the receiver clock. Depending upon the magnitude of the jitter in the receiver's clock, it may not be possible to produce an NTSC compliant color burst for video display. In addition, this jitter will at a minimum require the audio and video compressed data buffers to be larger than necessary and at worst cause the buffers to underflow or overflow. The side effects would include video freezing, possible audio chirps and the display colors changing dynamically due to the variations in the color burst.

Some networks, such as 1394 networks, provide isochronous capability that can eliminate this problem. Additionally, various methods, such as SRTS, have also been developed for ATM networks to address these timing issues. The SRTS method assumes that the transmitting device and the receiving device both have

access to a network clock for the synchronization. However, such methods may not be completely suitable for correcting jitter in an asynchronous collision detect network, such as an Ethernet network. Therefore, what is desired is a system and a method for synchronizing the clocks and controlling the data flow in an asynchronous network, and in particular, in an asynchronous collision detect network to overcome the above-noted problems. Furthermore, what is needed is a system and a method for synchronizing the clocks and controlling the data flow in an asynchronous network, wherein the broadband receiver is able to receive signals from one of a plurality of headend networks and each headend has its own system clock.

The present invention overcomes the jitter problems discussed above, and in particular, provides a system and a method for distributing multi-media content over an asynchronous network, wherein the receiver is able to receive inputs from multiple headend networks. More specifically, the present invention uses time stamps and clock adjustment factors to minimize and reduce the jitter effects caused by software processing and collisions on an asynchronous home network. A set of time stamps are used to perform synchronization of the clock in the client device to the headend clock and another set of time stamps are used to control data flow in the client device.

One set of time stamps is added for flow control. When a packet arrives from a broadband network, a sample of a local counter in the server is latched into a register. The local counter is clocked with a system clock of the server. The server attaches the sampled counter value to the received transport packets. Once a packet is time stamped, the server may place the packet in a transmit buffer. This time stamp indicates the moment at which the data was received from the broadband connection and placed into a buffer, and as such, can be used by the client device to remove the data from a receive buffer at the exact same rate as it entered the buffer.

Another set of time stamps, along with a clock adjustment factor, are used for synchronizing the client device clock with the head end clock. Each headend transmits synchronizing information, SCR, along with the multi-media data. When the server detects a SCR, the server runs an algorithm to calculate a normalization, or scaling factor, to be applied to the time stamps. The local clock of the server is not adjusted. The normalization allows the server's local clock to be expressed in terms

of the headend clock. The adjustment factor is applied to the broadband receiver's time stamps to express them in terms of the headend clock. At the moment a frame of data is being placed onto the network, a local counter running on the system clock is sampled. A time stamp based on the sampled clock is placed in the data frame in real-time at the physical layer of the network so that there is a fixed and constant amount of time from the sampling of the local counter and the sampled count value being placed onto the network. If a collision condition occurs, the time stamp is updated when the data is retransmitted. At the moment the new data frame is received in the physical layer of a client device, a sample of a local counter running on the client's local clock is latched into a register. This "receive-time" time stamp is then attached to the received frame of data and passed onto successive layers of the client's network protocol hardware and software. After the client data is passed through the communication protocols, the data is ready to be used by the client.

The client device uses the time stamps attached at the physical layer of the server's network interface and the clock adjustment factor, along with the time stamps that were attached when the data was received by the client device, to lock the client's local system clock to the headend system clock.

The time stamps attached to the data when the data arrived at the server from the broadband network, along with the clock adjustment factor, may then be used to determine the exact time when each data packet should be extracted from the client's buffer and passed on to the audio/video subsystems in the client. In this way the packet arrival timing at the server is exactly matched at the input to the A/V subsystem in the client device.

The invention is described with reference to the drawings, wherein:

Fig. 1 shows an asynchronous multi-media network including a media server coupled to a plurality of client devices;

Fig. 2 is a block diagram illustrating the elements of an asynchronous multi-media network including time stamps placed into the data at various points of the network according to the present invention;

Fig. 3 shows format of a data frame including the time stamps and clock adjustment factor utilized in the asynchronous multi-media network according to the present invention;

Fig. 4 is a block diagram illustrating the element of a broadband receiver adapted to receive satellite signals;

Fig. 5 is a block diagram illustrating the elements of the PCI DBS tuner board illustrated in Fig. 4;

Fig. 6 is a block diagram illustrating the elements of the client device illustrated in Fig. 2;

Fig. 7 is a block diagram illustrating the elements of the transport formatter board illustrated in Fig. 6;

Fig. 8 is a block diagram illustrating the elements of the client video decoder illustrated in Fig. 6;

Fig. 9 is a block diagram illustrating the time stamping board used in the asynchronous multi-media network according to the present invention;

Fig. 10 is a block diagram illustrating the time stamp controller in the time stamping board shown in Fig. 9;

Fig. 11 is a block diagram illustrating the receiver unit in the time stamp controller shown in Fig. 10; and

Fig. 12 is a block diagram illustrating the transmitting unit in the time stamp controller shown in Fig. 10.

Fig. 2 illustrates a digital home network utilizing the time stamps according to the present invention. The broadband input signal is provided by a headend (not shown) and received by broadband receiver/multi-media server 202. Server 202 may be configured to receive the input from one of a plurality of broadband sources, such as, satellite, cable or DSL, as indicated by input Sys 1, Sys 2... Sys n. This input signal may be in the form of transport packets, which include System Clock Recovery (SCRs) information used by server 202 to correlate its local system clock to the particular head end system clock. The SCRs are used to mange buffer levels and to derive video timing and color burst signals in an integrated receiver decoder (IRD). It is important to note that the SCR information and packet timing relationships must be maintained throughout system 200, from entry into the network to the point of final presentation, in order for system 200 to operate properly.

In accordance with the present invention, in order to ensure that the exact arrival time of packets at the client device may be reproduced, each packet arriving

from the broadband network is stamped with a time stamp, T1, when server 202 receives the packet. Time stamp T1 is a sample of a local counter that is clocked with a local system clock of server 202. Time stamp T1 and the packet are stored together in buffer 205 prior to being broadcast together over the DHN.

At some later point in time, a group of transport packets, along with header information, is assembled to form a data frame for transmission on the network DHN. The data frame includes portions where various time stamps and a clock adjustment factor may be placed. The moment the data frame starts to be placed onto the network, local counter in server 202, running on the local system time clock, is sampled to generate a counter sample time stamp T2. Time stamp T2 is placed in the outgoing data frame such that the time from sampling the counter value to the time the counter value enters the network is fixed and constant. It is important that this time is fixed and constant, and for this reason the counter sampling and time stamp placement is performed in the physical layer hardware of the network interface. Since the propagation delay from server 202 to client device 204 over the network is fixed and constant, the overall propagation delay from sampling T2 until the packet containing time stamp T2 arrives at client device 204 is also fixed and constant. Additionally, in a system where the symbol times may change, the present invention is able to provide the necessary synchronization as long as the propagation time in the network remains constant.

In the present invention, the system clock of server 202 is not changed. Instead, a clock adjustment factor is calculated and placed in the data frames to allow the time stamps to be used to frequency lock a client device clock to the particular headend clock. A suitable normalization factor may be, for example, clock adjust = SCR(0) - SCR(1) - [local clock (0) - local clock (1)]/(SCR(0)-SCR(1)). In this manner, the client device is able to frequency lock to any one of a plurality of network headends. Time stamp T2 and the clock adjustment factor are included in portions 306 and 305 in data frame 300 as shown in Fig. 3.

If a collision event occurs, time stamp T2 is updated the next time server 202 attempts to place the data frame onto the asynchronous network. It is important to note that the time stamp T2 is based on the local clock of server 202, which is not adjusted.

At the moment the data frame arrives at client device 204, a local counter of client device 204, running on the client system clock, is sampled to generate a time stamp T3. This counter value is compared to the adjusted time stamp T2 in the arriving data frame to ensure that the difference between the client clock and the headend clock is not changing. If this difference is changing over time, the client device will modify the voltage on its local VCXO to move its local system clock into frequency lock with the headend clock. For example,

error = (T2(0) - T2(1))*((clock adjust (0) + clock adjust (1))/2)-(T3(0)-T3(1)),

where (clock adjust (0) + clock adjust (1))/2 is the median clock adjust error, and error is how far off the client clock is from the headend clock.

Thus, time stamps T2 and T3, along with the adjustment factor, are used to frequency lock the system clocks of the headend and client device 204 together. Time stamps T2 and T3 may be discarded after they are used in client device 204 for synchronization. At this point, data frames that encapsulate transport packets and their associated time stamps T1 and the clock adjustment factor may be stored in buffer 203 in client device 204.

The next step is to remove data from the buffer of client device 204 at a rate that exactly matches the rate at which the packets arrived at the input of server 202. When client device 204 has accumulated enough data in a receive buffer to absorb a reasonable amount of network jitter client device 204 starts extracting data from the receive buffer, and making that data available to the decoders within client device 204. A local counter, which is based on the client clock is initialized with the count value stored with the first transport packet to be extracted from memory. The counter is clocked with the client's system clock, which is locked to the headend clock, so the subsequent packets may be extracted from memory when their stored time stamp value matches the counter value. In this manner, the rate of removal from the receive buffer of client device 204 matches the rate at which the data was received at server 202. The format of the data frame and placement of the time stamps T1, T2, T3 and the clock adjustment factor within the data frame are shown in fig. 3, wherein the time stamps T1 are generated as the packetized data is received and time

stamps T2 are placed into the data frames as the frames are placed on the network DHN.

The format of the data packets and the placement of the times stamps into the data frames in an Ethernet environment are further described below. The Ethernet packets in the present embodiment use the UDP/IP protocols, but it is to be understood that other suitable protocols, such as TCP, or "raw" Ethernet may be used to implement the time stamping features of the present invention. In accordance with the present invention, the IP header utilizes an IP option identifying the data frames that need to have a time stamp placed therein. The time stamps are placed in the data frames following the UDP header. There is a unique location in the extended UDP header for the outgoing and incoming packets.

The clock synchronization time stamps T2 and T3 and the clock adjustment factor are placed after the UDP header. Time stamp T2 is a 32-bit time stamp and is applied when the IP packet has an option 25 set. The time stamp is placed into the data frame at the physical network layer. Although the present embodiment utilizes option 25 to indicate the presence of time stamps in the frame, it is to be understood that other suitable method for indicating time stamps, such as designating specific IP address and port number may be utilized.

Time stamp T3 is also a 32-bit time stamp and is applied when an IP packet with option 25 is detected. This time stamp is also placed into the data frame at the physical network layer.

The data flow control time stamps T1 are placed into the data stream when the data is received from a broadband network. Time stamps T1 are generated using a local counter in server 202. By placing a time stamp T1 on each transport packet as the packet arrives at server 202, client device 204 can reproduce the same bit rate as the rate received at server 202. Data flow control time stamps T1 are in the UDP payload. Table 1 shows the data format used in the exemplary embodiment of the present invention.

Table 1 - Data including transport packets and time stamps

0 15 16 31
Transport Packet Marker
16-bit MSBs of time stamp T1(0)
16-bit LSBs of time stamp T1(0)
Transport Packet 0
Transport Packet Marker
16-bit MSBs of time stamp T1(1)
16-bit LSBs of time stamp T1(1)
Transport Packet 1
. .
Transport Packet Marker
16-bit MSBs of time stamp T1(n)
16-bit LSBs of time stamp T1(n)
Transport Packet n

Once the data frame has been time stamped, the UDP checksum is checked. If the checksum is not zero, the old checksum is replaced with a newly created checksum. After resolving the UDP checksum, a new CRC is calculated and replaces the old CRC value in the MAC layer. If at any point of the packet interrogation it is determined that the data frame is not a valid time stamp frame, the frame is passed through with no alterations to the data.

The time stamp value is generated from an external system clock. The external clock drives a 32-bit counter. A snapshot of the counter is taken when a data frame is placed onto the network. On an incoming packet, a snapshot of a counter in the client device is taken. If it is a valid time stamped packet, the snapshot of the counter is placed into the payload portion of the data frame. The 32-bit counter is a free running rollover counter synchronous to the external system clock.

The following describes the search sequence to determine whether a particular data frame needs a time stamp, and if so, where the time stamp data is placed. The first test is to validate the MAC frame. Validating the MAC frame

requires the CRC to be calculated and compared. If the CRC test fails, the frame is allowed to pass through without any modifications to ensure that a valid CRC is not added to a data frame that was received with an invalid CRC. If the CRC test is passed, a second test is conducted to find an Internet Protocol (IP) in the MAC frame. Twenty octets into the MAC frame, from the 802.3 specification section 3.2.6, Length/Type, is the Length/Type field. From ITEF RFC 1700 EtherTypes table, an Internet IP (Ipv4) is 0x800 (Hexadecimal). If the value is not 0x800, the packet is passed through. If the Length/Type is 0x800, another test is conducted to determine the size of the IP header and whether it contains an UDP packet. The contents of a MAC frame are specified in IEEE 802.3, section 3.1.1.

If the MAC frame contains an IP packet, the next test determines the size and the option settings. First, in order to determine the size, the second nibble of the IP packet, which is the Header Length, is examined. The Header Length is the number of 32-bit words that make up the IP header. Without options, the IP header is normally 20 bytes, so the Header Length is 5. The IP Header options have a 20 decimal offset from the start of the IP Header.

The steps to determine whether the packet is a valid time stamp packet is now described. If any of the steps fail, the packet is passed through without any modifications. First, the header length field is examined to determine whether it is greater than 5. If so, the 1st IP option located 20 bytes from the beginning of the IP Header is checked against 0x99040000. The format of the option is shown in Table 2 below:

Table 2 IP Option 25

Copy (1 bit)	Class (2 bits)	Number (5 bits)	Length (8 bits)	Parm (16 bits)
1	0	25	4	0

Table 3 shows an IP header with IP option 25 included.

Table 3 IP Header with IP Option 25

0		15 16		31
4-bit version	4-bit header length	8-bit type of service (TOS)	16-bit total length (in bytes)	
16-bit identification			3-bit flag	13-bit fragment offset
8-bit time to live (TTL)	8-bit protocol		16-bit header checksum	
32-bit source IP address				
32-bit destination IP address				
32-bit option 25				
data				

If the IP options include the designated time stamping option described above, the present system places a time stamp into the data frame. The time stamps are placed in the UDP payload portion that follows the IP header, and in particular, at the beginning of the payload portion. The UDP header is 8 bytes long. The first 32 bit word of the UDP payload is reserved for time stamp T3, and the second 32 bit word of the UDP payload is reserved for T2.

A significant aspect of the present invention is that the time stamps and the CRC are generated and placed into the data frames at the physical layer. As such, the time stamps and CRC are placed into the data frames as the data frames are actually being placed onto the network or are received from the network. This allows the time stamps to accurately reflect the time the data enters the network and the time the data is received from the network.

The elements of a system for implementing the time stamping described above is now described in further detail with respect to a DBS receiver. However, it is to be understood that similar time stamping may be performed for multi-media data received on other types of broadband networks, such as DSL or cable.

As shown in Figure 4, server 202 consists of three functional blocks connected by a bus. The satellite signal is received by server 202 through PCI DBS tuner 402. Host controller 406 reads the transport packets from the satellite signal out of PCI DBS tuner 402 via PCI bus interface 405. Host controller 406 then processes the data and sends the processed data out to time stamp board 404 via well-known protocols, including UDP. Also, PCI DBS tuner 402 and time stamp board 404 have a common clock between them. The clock is generated on PCI DBS tuner 402, and used as the time reference for time stamp board 404.

In operation, host controller 406 reads the transport packets out of the FIFO memory of PCI DBS tuner 402. The packets are then processed according to the network protocol software and the network frames are then written into the Ethernet MAC of time stamp board 404. The MAC waits for the Ethernet Network to become available, and then sends the data through a time stamp controller to an Ethernet PHY. When a video network packet is detected by the time stamp controller, a time stamp is added to that packet. The time stamp T2 is a sample of a counter in the time stamp controller that is clocked by a clock in server 202. In server 202, the clock source for the time stamp controller is a VCXO in PCI DBS tuner 402.

Figure 5 shows a block diagram of PCI DBS tuner 402. PCI DBS tuner 402 performs several functions. First, PCI DBS tuner 402 ensures that the correct satellite transponder is selected, tuned, and demodulated. This is accomplished by satellite tuner 502 and demodulator 504. These devices perform the well-known conversion of a satellite signal into a digital transport stream.

Second, PCI DBS tuner 402 filters the packets of the transport stream. Tuner controller 506 contains logic to filter out only the transport packets requested by host controller 406. The transport packets are then stored in FIFO buffer 512. Host controller 506 can then read the packets out of FIFO buffer 512 via PCI bus interface 405. In order to facilitate the PCI bus interface, a PCI bridge chip 508 is used.

Additionally, time stamps T1 can be added to each transport packet using the system clock from VCXO 510. Time stamp T1 is based on a sample of the local counter at the moment the packet is received at tuner 402.

The elements of network client 204 are shown in Figure 6. Data arrives at the client's time stamp board 602, and time stamp T3 is added to the data frame at the

moment of arrival. The packets are processed by client controller 608, which is connected to time stamp board 602 via PCI bus 605. The transport packets are extracted from the network frames, and are then written to transport formatter board 604. The output of transport formatter 604 is a serial transport stream that may be used by decoder 606. The output of decoder 606 is connected to a television or other display device. Figure 8 shows the elements of decoder 606.

Client controller 608 is responsible for executing a clock recovery algorithm. The data frame departure times T2 from server 202, adjusted by the clock adjustment factor, are compared to the arrival times T3 at client 204. In this manner, client controller 608 can determine whether decoder VCXO 812 is faster or slower than the headend clock. Client controller 608 sends commands to decoder controller 810 to speed up or slow down decoder VCXO 812 through a low speed serial (RS-232) connection.

The time stamp board of the client device is similar to the time stamp board of the server device except that the recovered clock source in the client device is derived from the clock of decoder 606.

The elements of transport formatter 604 are shown in Fig. 7. Transport formatter 604 is connected to client controller 608 via PCI bus 702. A PCI bus interface converts the PCI bus to a standard local bus, which is connected to transport formatter controller 704. Transport formatter controller 704 outputs a serial transport stream at TTL logic levels. High speed serial converter 706 circuit converts those signals to low voltage differential signals, which are passed to decoder 606.

Transport formatter controller 704 has two modes of operation. The first is flow control mode in which transport packets are forwarded to decoder 606 according to the optional time stamps added by tuner controller 506 of PCI DBS tuner 402. In this manner, the relative transport packet departure times from transport formatter 604 match the transport packet arrival times at tuner controller 506. This has the effect of making the packet timing at the input of decoder 606 appear to be connected directly to tuner 502 and demodulator 504 of PCI DBS tuner 402.

The second mode of operation allows the data to flow immediately from transport formatter controller 704 to decoder 606. In this mode, decoder 606 is

required to buffer the data internally. This requires decoder 606 to contain more memory than the first mode.

Data arrives from transport formatter 604 into high speed serial interface 802 of decoder 606. Transport processor 804 sorts the incoming transport packets and puts them into the correct buffer (video, audio, program guide, other data, etc.). The video and audio data are decompressed in the video/audio decoder 806. After that, decompressed digital data is converted to analog NTSC signals via NTSC encoder/audio DAC 808.

The time stamping board used in implementing the present time stamping method is now further described. Although, described with reference to the time stamping board associated with server 202, as noted above, the time stamping board associated with client 204 is similar. Time stamping board 404 is a PCI Ethernet NIC that can support both 10Mb/s and 100 Mb/s data rates. It is capable of resetting the UDP check sum and placing 32-bit time stamps in the data frames, as well as recalculating the CRC for the designated data frames. The external interfaces allow board 404 to be integrated into PC PCI or embedded PCI architectures. An external clock interface is provided to generate time stamp values.

Fig. 9 illustrates a block diagram of time stamping board 404. The time stamping board comprises an input jack 904, for example RJ45, coupled to Ethernet network 902. Input jack 904 is coupled to 10/100 Base-TX Phy transformer 906. Transformer 906 is coupled to physical layer device 908, which is a full feature physical layer device with integrated PMD sublayers to support both 10BASE-T and 100BASE-X Ethernet protocols. Suitable physical layer device 908 includes, but is not limited to DP83843VJE manufactured by National Semiconductor Corporation. Physical layer device 908 is coupled to bus buffer 912, which translates the voltage levels between physical layer device 908 and time stamp controller 916. Bus buffer 912 is a low voltage CMOS Octal Bus buffer. It is ideal for low power and high-speed 3.3V applications and can be interfaced to 5V-signal environment for both inputs and outputs. Bus buffer 912 is coupled to time stamp controller 916, which controls and performs much of the time stamping functions, including, resetting the UDP check sum, placing the time stamps and recalculating the new CRC for the valid data frames in both the receive and transmit directions.

Time stamp controller 916 is coupled to configuration device 918, which stores and loads the program to operate time stamp controller 916. Time stamp controller 916 is also coupled to Ethernet MAC unit 914, which is a LAN controller for both 10Mb/s and 100Mb/s data rates. Ethernet MAC 914 provides a direct interface to the peripheral component interconnect (PCI) local bus or the CardBus.

Time stamp controller 916 provides the MII interfaces to the PHY and the MAC. As shown in Fig. 10, controller 916 consists of receiver unit 1002 and transmitter unit 1004. The two units are implemented separately, but share the same external system clock and the reset signal.

As shown in Fig. 11, receiver unit 1002 comprises receiver controller 1102, time stamp generator 1114, shift register 1112, verify CRC module 1110, calculate new CRC module 1106, 3:1 MUX 1108 and 4:1 MUX 1104. Receiver controller 1102 detects the status of data stream to enable the 3:1 MUX and the 4:1 MUX for appropriate out put data (receive data, reset UDP check sum, time stamp, new CRC).

Calculate new CRC module 1106 calculates the CRC check sum for the modified data stream (Ethernet packet which has time stamp and reset UDP check sum), this new CRC will be placed into the modified data stream before it arrives at the MAC. The difference of receiver unit 1002 to transmitter unit 1004 is that receiver unit 1002 included verify CRC module 1110. It is necessary to verify the CRC on packets that are received, due to the fact that this data may have been corrupted on the medium. If the CRC is not correct, this indicates that the data has indeed become corrupted. Instead of adding a new, correct CRC to the data, the data is allowed to pass through to the MAC with an incorrect CRC.

Controller 1102 synchronizes and controls the operations of all modules and components. Two clocks are used: a receive clock and an external system clock. The time stamp is generated by the external clock and latched into the receive clock domain at the start of every packet. Data is received from the PHY and the appropriate data (receive data, reset UDP check sum, time stamp, new CRC) is transmitted to the MAC in accordance with IEEE 802.3.

When the data is received, the time stamp T3 is placed in the appropriate area and a new CRC is calculated and replaces the old CRC value in the MAC layer. Again, a significant aspect of the present invention is that the time stamps and the

CRC are generated and placed in the data frames at the physical layer as the data frame is received from the network. As such, the time stamps and the CRC, which reflect the actual time the data frames are received from the network, are quickly and efficiently placed in the data frames as they move to the MAC layer. If at any point of the packet interrogation it is determined that the data frame is not a valid time stamp frame or there are errors while receiving, the data frame is allowed to pass through with no alterations to the data.

As shown in Fig. 12, transmitter unit 1202 comprises transmitter controller 1202, time stamp generator 1210, shift register 1208, calculate new CRC module 1206, 3:1 MUX 1212 and 4:1 MUX 1204. Transmitter controller 1202 also controls the operations of all modules and components for each appropriate process. Two clocks are used: a transmit clock and an external system clock. The time stamps are generated by the external clock and latched into the receive clock domain at the start of every packet. Here again, the time stamps are generated and placed into the data frames at the physical layer.

While this invention has been described as having a preferred design, the present invention can be further modified within the spirit and scope of this disclosure. For example, the time stamping features may be implemented in other asynchronous networks, including, but not limited to, HPNA, PLC, and certain wireless networks, for example IEEE 802.11b. This application is therefore intended to cover any variations, uses, or adaptations of the invention using its general principles. Further, this application is intended to cover such departures from the present disclosure as come within known or customary practice in the art to which this invention pertains and which fall within the limits of the appended claims.

CLAIMS

1. A server apparatus for receiving packetized multi-media data and transmitting the packetized multi-media data over an asynchronous network, the apparatus comprising:

an input for receiving packetized multi-media data from a selected one of a plurality of signal sources, the packetized multi-media data including synchronizing information associated with the selected signal source;

a clock;

a controller, coupled to the input and the clock, for generating a clock adjustment factor for correlating the clock to a clock associated with the selected signal source in response to the synchronizing information, the controller generating data frames having the packetized multi-media data and the clock adjustment factor included therein;

an output, coupled to the asynchronous network, for transmitting the data frames to a client device;

a time stamping device, coupled to the controller, the clock and the output, for placing a synchronizing time stamp in the each data frame, the synchronizing time stamp being indicative of the time the respective data frame is placed onto the asynchronous network, whereby the synchronizing time stamp and the clock adjustment factor may be used by a client device for synchronizing a clock associated with the client device with the clock associated with the selected signal source.

2. The server apparatus according to claim 1, wherein the controller generates data flow time stamps in response to the receipt of the packetized multi-media data, each data flow time stamp being indicative of a time a respective multi-media data packet is received at the input, the data frame including the packetized multi-media and associated data flow time stamps, wherein the data flow time stamps and the clock adjustment factor may be used by the client device to control the data flow rate in the client device to correspond to a rate that the packetized multi-media data is received at the input.

3. The server apparatus according to claim 2, wherein the data frame comprises a data frame header, time stamp data portion for accepting synchronizing time stamps, a clock adjustment portion for accepting the clock adjustment factor, and a data portion for accepting packetized multi-media data and their corresponding data flow time stamps.

4. The server apparatus according to claim 3, wherein the time stamp data portion includes a first time stamp data portion for accepting the synchronizing time stamp and a second time stamp data portion for accepting a second synchronizing time stamp added by the client device.

5. The server apparatus according to claim 2, wherein the asynchronous network is an Ethernet network.

6. The server apparatus according to claim 2, wherein the asynchronous network is an HPNA network.

7. The server apparatus according to claim 2, wherein the asynchronous network is a PLC network.

8. The server apparatus according to claim 2, wherein the asynchronous network is a wireless network.

9. A method for transmitting packetized multi-media data over an asynchronous network, the method comprising:

receiving the packetized multi-media data from a selected one of a plurality of signal sources, the packetized multi-media data having synchronizing information associated with the selected signal source;

generating a clock adjustment factor for correlating a server clock to a clock associated with the selected signal source in response to the synchronizing information;

generating data frames having the packetized multi-media data and the clock adjustment factor included therein;

placing the data frames on the asynchronous network to transmit the data frames to a client device, wherein a synchronizing time stamp is generated and placed in each data frame, each synchronizing time stamp being indicative of the time the respective data frame is place in the asynchronous network, wherein the synchronizing time stamps and the clock adjustment factor may be used by the client devices to synchronize a clock associated with the client device to the clock associated with the selected signal source.

10. The method according to claim 9, further comprising the steps of generating data flow time stamps upon receipt of the packetized multi-media data, each data flow time stamp being indicative of a time a corresponding multi-media data packet is received, and the step of generating data frames includes generating data frames having the packetized multi-media data and the data flow time stamps included therein, whereby the client device can control the data flow rate in the client device to correspond to the rate the packetized multi-media data is received in response to the data flow time stamps and the clock adjustment factor.

11. An apparatus for processing packetized multi-media data received over an asynchronous network, the packetized multi-media data comprising data frames having synchronizing time stamps indicative of the time respective data frames are placed on the asynchronous network, and a clock adjustment factor that correlates a server clock with a clock associated with the signal source, the apparatus comprising;

an input, coupled to the asynchronous network, for receiving the data frames transmitted by the server device operatively coupled to a selected one of a plurality of signal sources;

a time stamp device, coupled to the input, for generating a second synchronizing time stamp indicative of the time the data frame is received at the input;

a clock;

a controller, coupled to the clock and the time stamp device, for synchronizing the clock to a clock associated with the selected one of a plurality of signal sources in response to the synchronizing time stamps, the clock adjustment factor and the second synchronizing time stamps.

12. The apparatus according to claim 11, wherein the data frames further include data flow time stamps associated with the packetized multi-media data, the data flow time stamps being indicative of the time each multi-media data packet is received by the server, and further comprising a buffer for storing the packetized multi-media data, and a decoder, wherein the controller controls the rate that the packetized multi-media data is transferred to the decoder to correspond to the rate that the packetized multi-media data is received at the server in response to the data flow time stamps and the clock adjustment factor.

13. The apparatus according to claim 12, wherein the controller generates an error signal indicative of the difference between the clock and the clock associated with the signal source in response to the synchronizing time stamps, the clock adjustment factors, and the second synchronizing time stamps, and frequency locks the clock to the signal source clock in response to the error signal.

14. The apparatus according to claim 12, wherein the asynchronous network is an Ethernet network.

15. The apparatus according to claim 12, wherein the asynchronous network is an HPNA network.

16. The apparatus according to claim 12, wherein the asynchronous network is a PLC network.

17. The apparatus according to claim 12, wherein the asynchronous network is a wireless network.

18. A method for controlling a client device in response to the receipt of data frames received over an asynchronous network, each data frames including packetized multi-media data, a synchronizing time stamp indicative of the time the data frame is placed on the asynchronous network, and a clock adjustment factor indicative of a correlation between a sever clock and a signal source clock, the method comprising;

receiving the data frames transmitted by a server device;

generating second synchronizing time stamps upon receipt of the data frames, each second time stamp being indicative of the time a respective data frame is received over the asynchronous network; and

synchronizing a clock in the client device to the signal source clock in response to the synchronizing time stamps, the clock adjustment factor and the second synchronizing time stamps.

19. The method according to claim 18, wherein the synchronizing step includes generating an error signal indicative of the difference between the client device clock and the signal source clock in response to the synchronizing time stamps, the clock adjustment factors, and the second synchronizing time stamps, and frequency locking the client device clock to the signal source clock in response to the error signal.

20. The method according to claim 18, wherein the data frames further include data flow time stamps, each data flow time stamp being indicative of a time a corresponding multi-media data packet is received by the server device, further comprising the step of:

controlling the rate at which the packetized multi-media data is transferred from a buffer to a decoder to correspond to the rate at which the packetized multi-media data is received at the server device in response to the data flow time stamps and the clock adjustment factor.

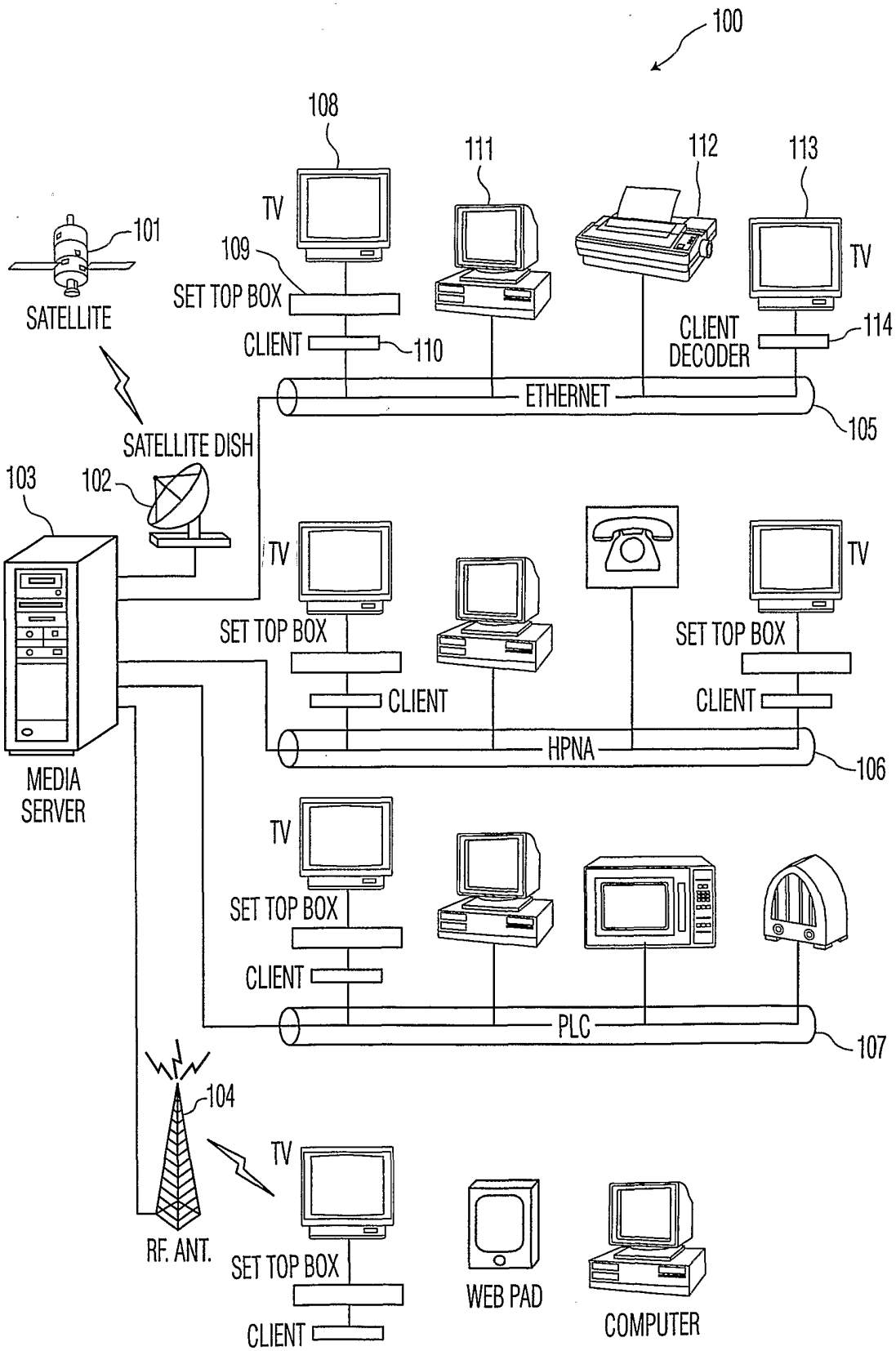

100
108
TV
109
SET TOP BOX
CLIENT
110
111
112
113
TV
CLIENT DECODER
114
ETHERNET
105
101
SATELLITE
SATELLITE DISH
103
102
MEDIA SERVER
TV
SET TOP BOX
CLIENT
HPNA
106
TV
SET TOP BOX
CLIENT
TV
SET TOP BOX
CLIENT
PLC
107
104
RF. ANT.
TV
SET TOP BOX
CLIENT
WEB PAD
COMPUTER

FIG. 1

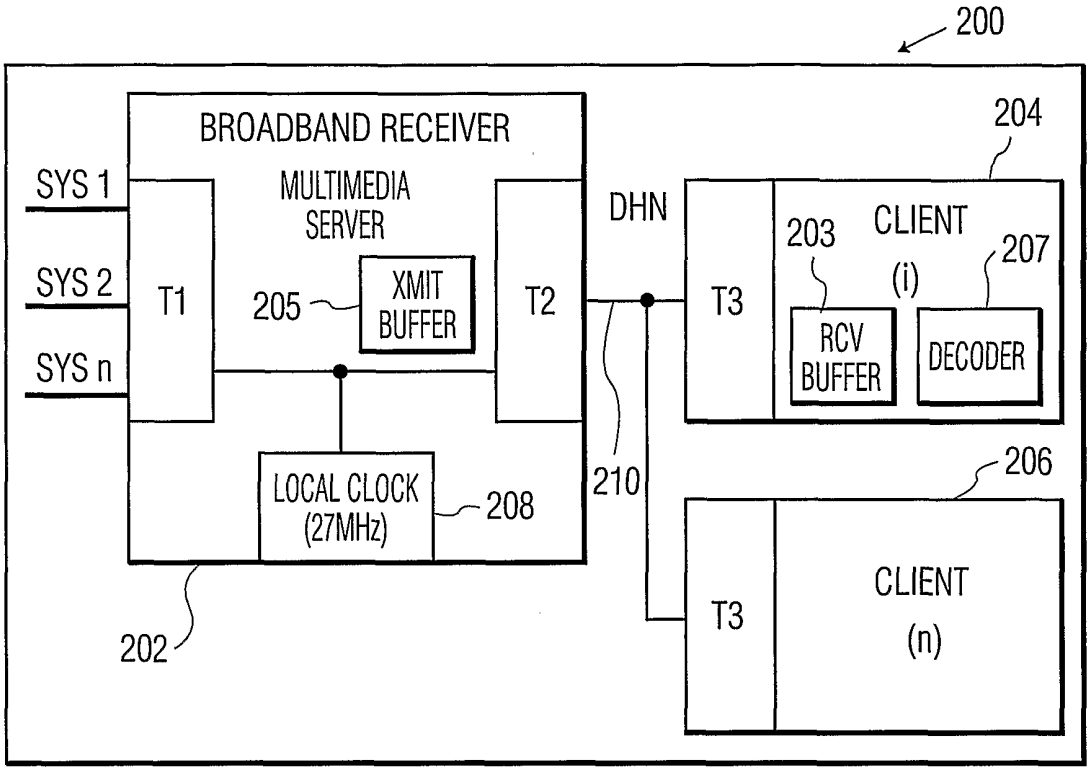

200
BROADBAND RECEIVER
SYS 1
SYS 2
SYS n
MULTIMEDIA SERVER
T1
205
XMIT BUFFER
T2
LOCAL CLOCK (27MHz)
208
202
DHN
210
204
T3
CLIENT (i)
203
RCV BUFFER
207
DECODER
206
T3
CLIENT (n)

FIG. 2

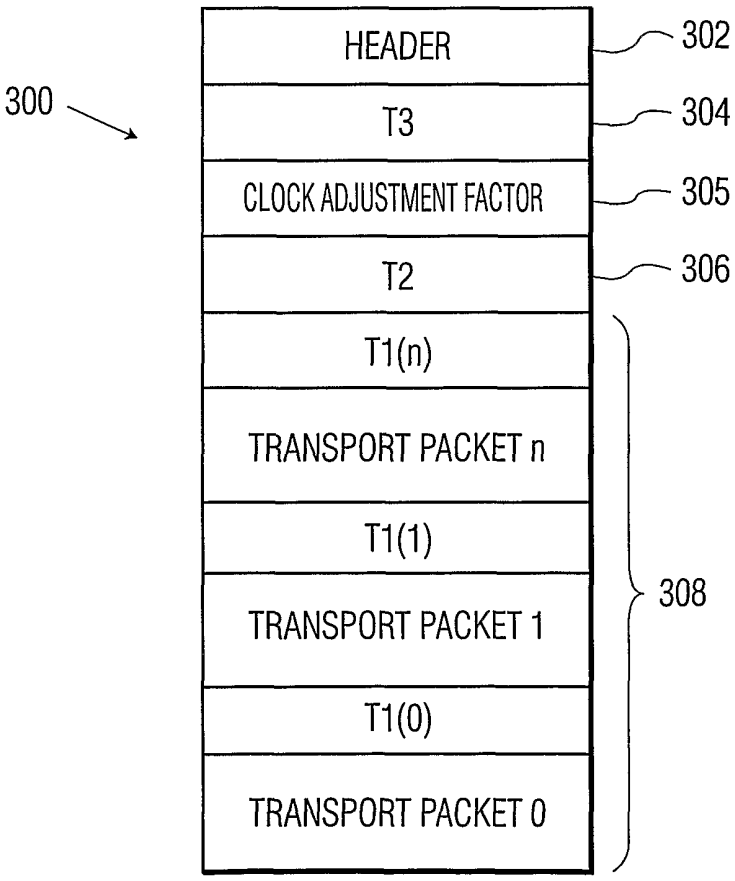

300
HEADER
302
T3
304
CLOCK ADJUSTMENT FACTOR
305
T2
306
T1(n)
TRANSPORT PACKET n
T1(1)
TRANSPORT PACKET 1
T1(0)
TRANSPORT PACKET 0
308

FIG. 3

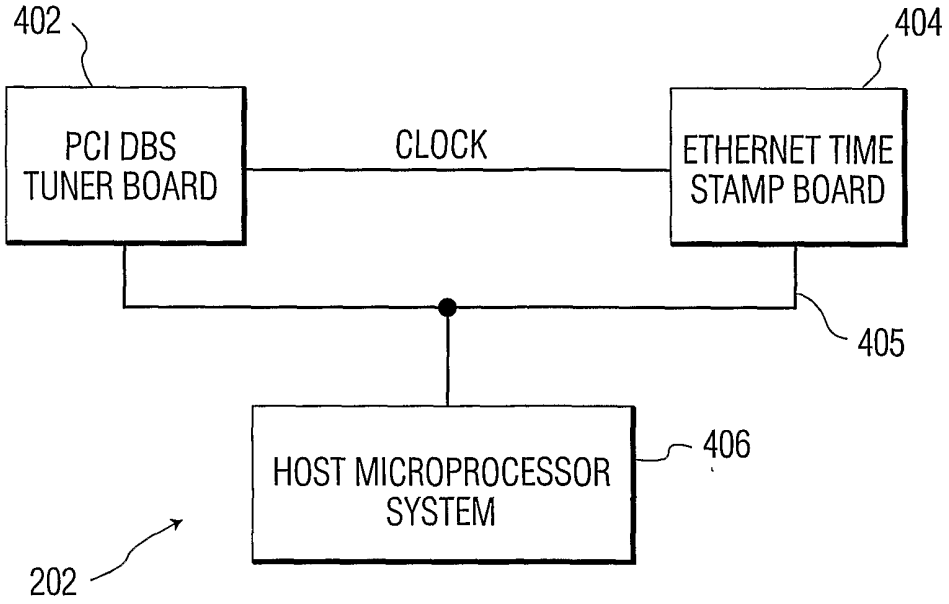

402
404
PCI DBS TUNER BOARD
CLOCK
ETHERNET TIME STAMP BOARD
405
HOST MICROPROCESSOR SYSTEM
406
202

FIG. 4

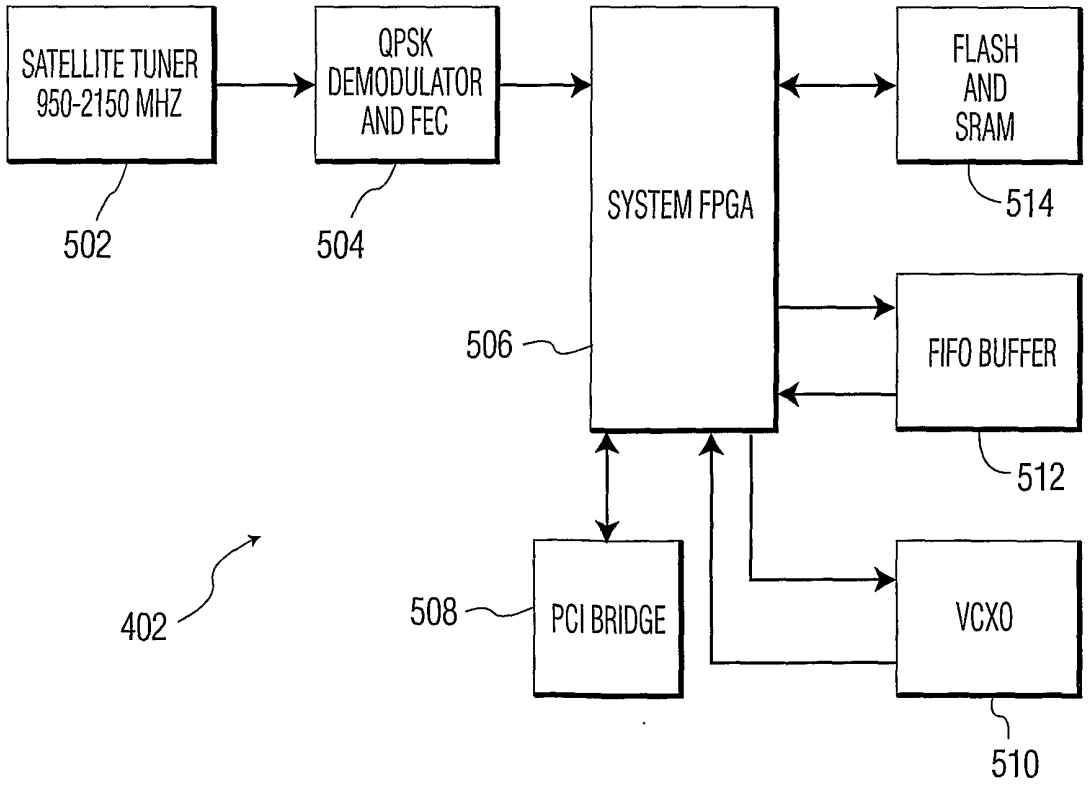

SATELLITE TUNER 950-2150 MHZ
QPSK DEMODULATOR AND FEC
SYSTEM FPGA
FLASH AND SRAM
514
502
504
506
FIFO BUFFER
512
402
508
PCI BRIDGE
VCXO
510

FIG. 5

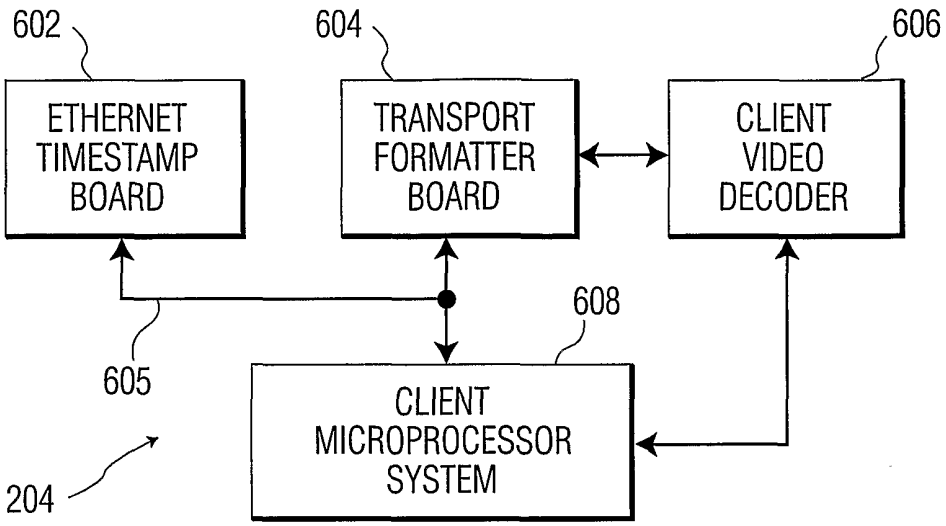


FIG. 6

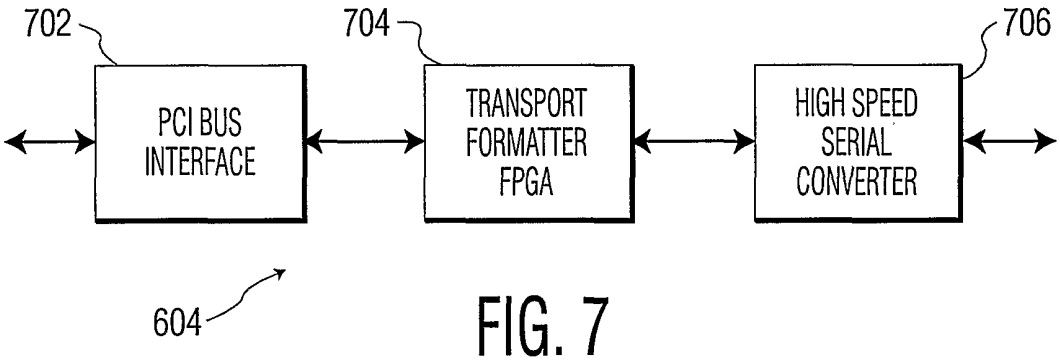


FIG. 7

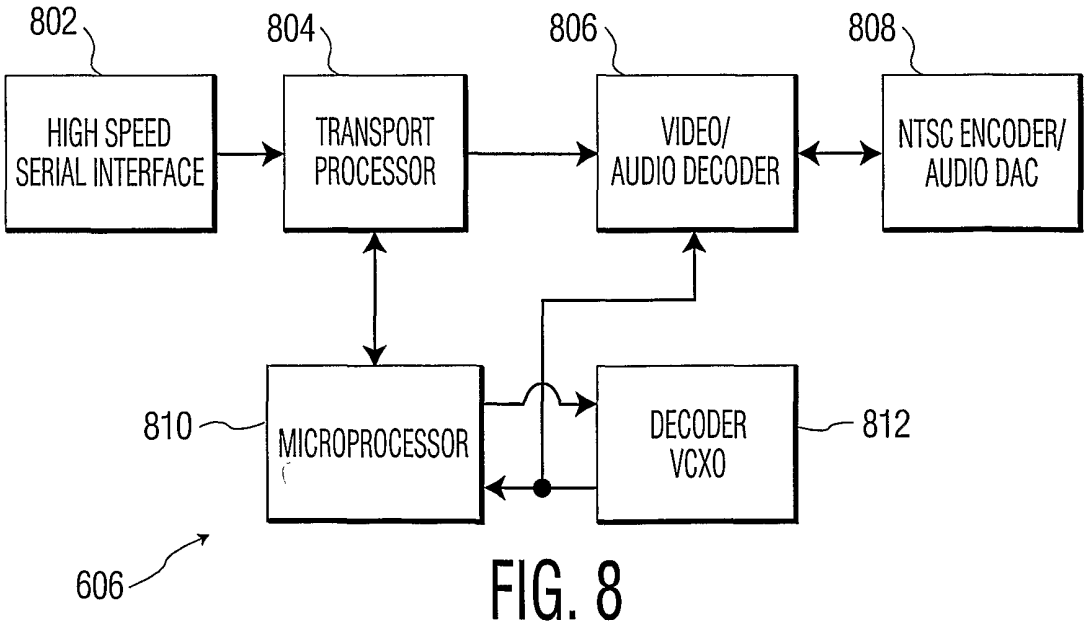


FIG. 8

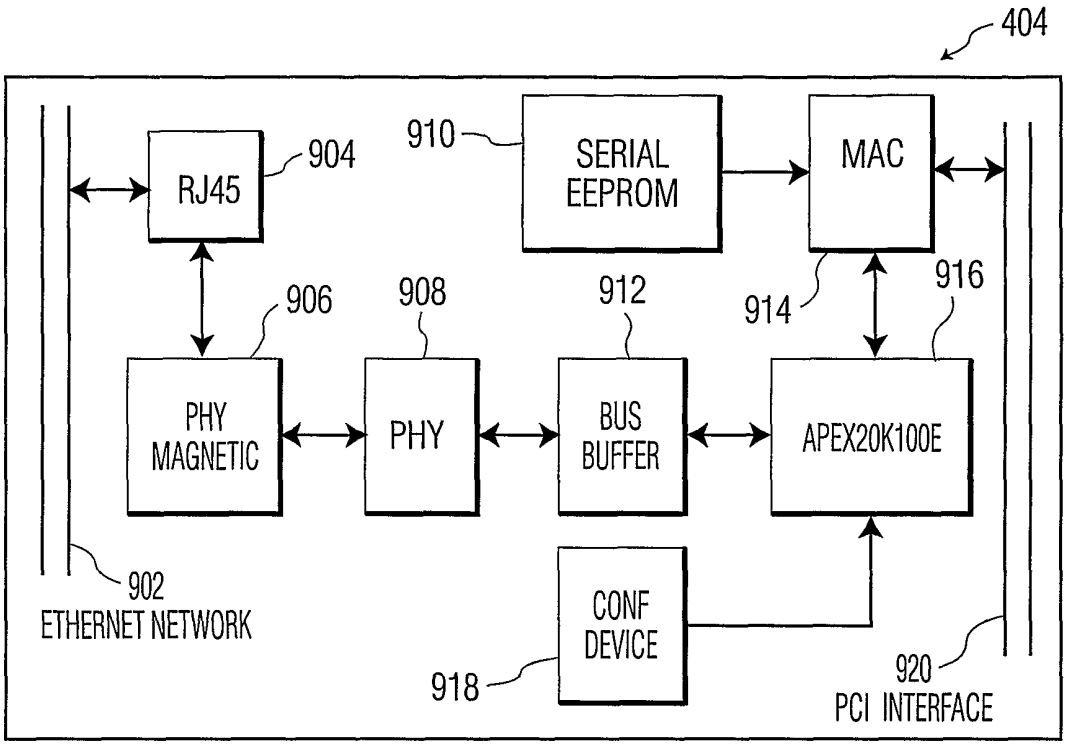

404
RJ45
904
910
SERIAL EEPROM
MAC
906
908
912
914
916
PHY MAGNETIC
PHY
BUS BUFFER
APEX20K100E
902
ETHERNET NETWORK
CONF DEVICE
918
920
PCI INTERFACE

FIG. 9

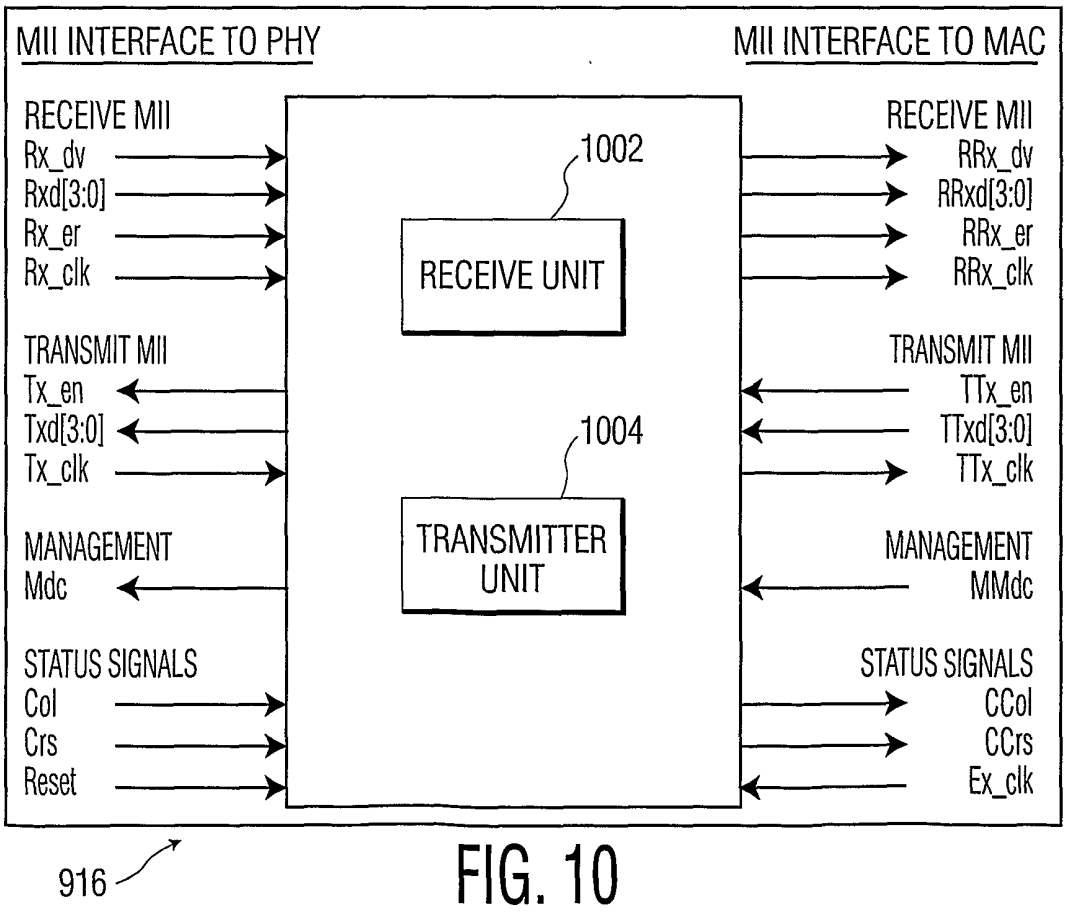

MII INTERFACE TO PHY
MII INTERFACE TO MAC
RECEIVE MII
Rx_dv
Rxd[3:0]
Rx_er
Rx_clk
1002
RECEIVE UNIT
RECEIVE MII
RRx_dv
RRxd[3:0]
RRx_er
RRx_clk
TRANSMIT MII
Tx_en
Txd[3:0]
Tx_clk
1004
TRANSMITTER UNIT
TRANSMIT MII
TTx_en
TTxd[3:0]
TTx_clk
MANAGEMENT
Mdc
MANAGEMENT
MMdc
STATUS SIGNALS
Col
Crs
Reset
STATUS SIGNALS
CCol
CCrs
Ex_clk
916

FIG. 10

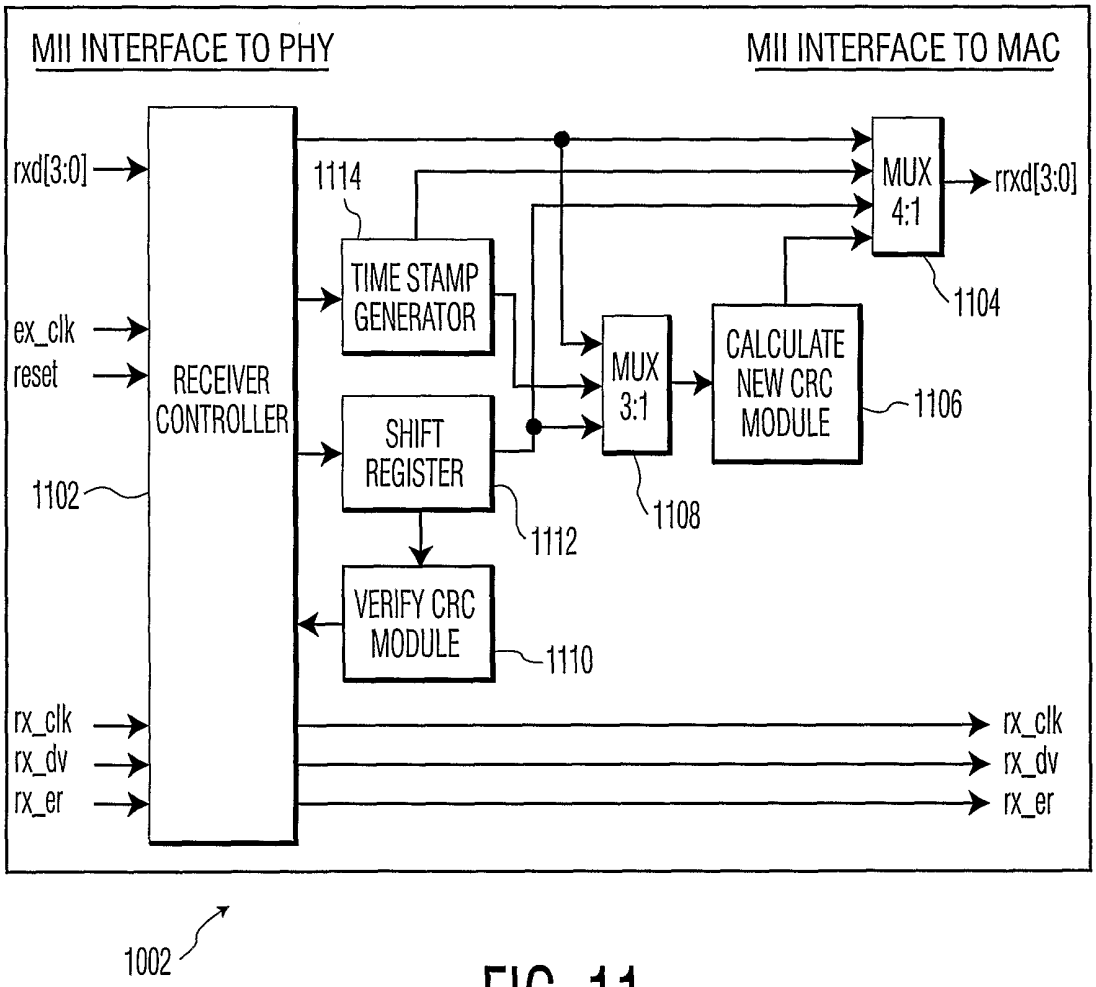

MII INTERFACE TO PHY
MII INTERFACE TO MAC
rxd[3:0]
ex_clk
reset
1102
rx_clk
rx_dv
rx_er
RECEIVER CONTROLLER
1114
TIME STAMP GENERATOR
SHIFT REGISTER
1112
VERIFY CRC MODULE
1110
MUX 3:1
1108
CALCULATE NEW CRC MODULE
1106
MUX 4:1
1104
rrxd[3:0]
rx_clk
rx_dv
rx_er
1002

FIG. 11

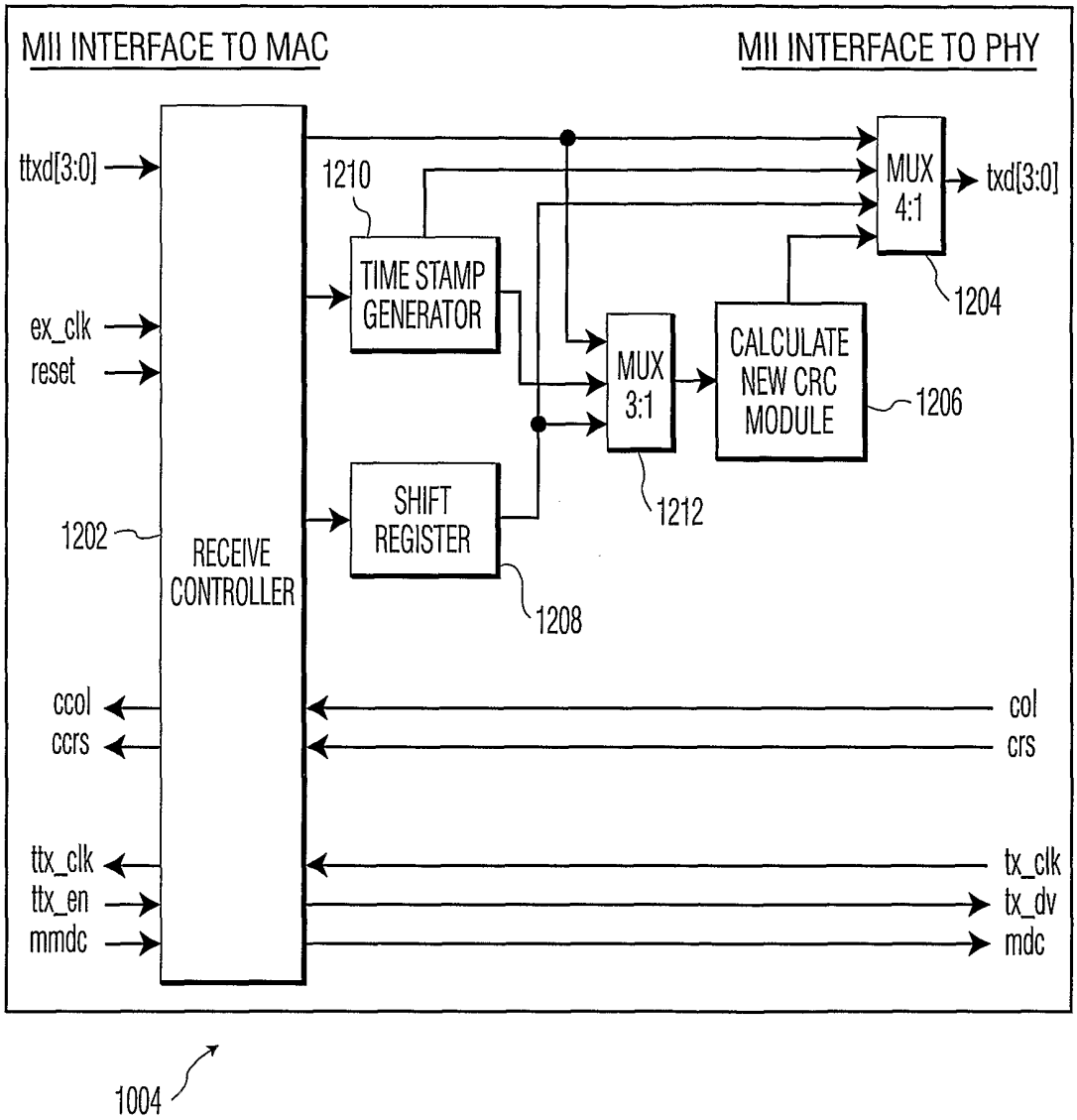

MII INTERFACE TO MAC
MII INTERFACE TO PHY
ttxd[3:0]
ex_clk
reset
1202
RECEIVE CONTROLLER
ccol
ccrs
ttx_clk
ttx_en
mmdc
1210
TIME STAMP GENERATOR
SHIFT REGISTER
1208
MUX 3:1
1212
CALCULATE NEW CRC MODULE
1206
MUX 4:1
1204
txd[3:0]
col
crs
tx_clk
tx_dv
mdc
1004

FIG. 12