US6091769: Video decoder having an interfacing function for picture synchronization

Video decoder with interfacing function for picture synchronization, has interfaces to multiplex extracted time stamp information with start code of corresponding picture to output encoded video data suitably

US United States of America

Moon, Heon-Hee; Suwon, Republic of Korea

Samsung Electronics Co., Ltd., Kanagawa, Japan

2000-07-18 / 1997-06-11

US1997000872734

1996-06-11 **KR1996000020792**

A video decoder has an interfacing function for picture synchronization which can synchronize a decoding and display operation in units the size of a picture. The video decoder includes a transport stream (TS) demultiplexer and an interfacer. The TS demultiplexer is a packet separator for separating packetized elementary stream (PES) packet data of a desired program from a received transport stream (TS). The interfacer has an input buffer for receiving and storing the PES packet data, a PES filter for extracting the encoded video data from the PES packet data stored in the input buffer, a presentation time stamp (PTS) and decoding time stamp (DTS) extractor for extracting the time stamp information of the PTS and DTS contained in the PES packet data stored in the input buffer, a picture start code (PSC) detector for detecting the PSC contained in the PES packet data stored in the input buffer, and a multiplexer for multiplexing the time stamp information extracted from the PTS/DTS extractor with the corresponding picture start code of the encoded video data output from the PES filter, based on the PSC detected in the PSC detector, and outputting the multiplexed result. Thus, the time stamp information can be decoded together with the bit steam video data, to thereby facilitate synchronization of a picture unit and enable normal decoding and display.

Sughrue, Mion, Zinn, Macpeak & Seas, PLLC ;

Lee, Richard;

Family:

US6091769	2000-07-18	1997-06-11	Video decoder having an interfacing function for picture synchronization
KR0203262B1	1999-06-15	1996-06-11	INTERFACE DEVICE OF VIDEO DECODER FOR SYNCRONIZATION OF PICTURE
HK1046611A1	2004-12-17	2002-11-06	Video decoding apparatus for decoding and displaying synchronously in a picture unit.
HK1046610A1	2004-12-17	2002-11-06	Video decoding apparatus for decoding and displaying synchronously in a picture unit.
HK1046609A1	2004-12-17	2002-11-06	Video decoding apparatus for decoding and displaying synchronously in a picture unit.
HK1046607A1	2004-12-17	2002-11-06	Video decoding apparatus for decoding and displaying synchronously in a picture unit.
HK1046606A1	2004-12-17	2002-11-06	Video decoding apparatus for decoding and displaying synchronously in a picture unit.
CN1370015A	2002-09-18	2001-10-23	VIDEO DECODER FOR SYCHRONOUS DECODING

			DISPLAYING USING IMAGE AS UNIT
CN1370014A	2002-09-18	2001-10-23	VIDEO DECODER FOR SYNCHRONOUS DECODING DISPLAYING USING IMAGE AS UNIT
CN1370013A	2002-09-18	2001-10-23	VIDEO DECODER FOR SYNCHRONOUS DECODING DISPLAYING USING IMAGE AS UNIT
CN1370012A	2002-09-18	2001-10-23	VIDEO DECODER FOR SYNCHRONOUS DECODING DISPLAYING USING IMAGE AS UNIT
CN1370011A	2002-09-18	2001-10-23	VIDEO DECODER FOR SYCHRONOUS DECODING DISPLAYING USING IMAGE AS UNIT
CN1370010A	2002-09-18	2001-10-23	VIDEO DECODER FOR SYNCHRONOUS DECODING DISPLAYING USING IMAGE AS UNIT
CN1171700A	1998-01-28	1997-06-10	VIDEO DECODER HAVING INTERFACING FUNCTION FOR PICTURE SYNCHRONIZATION
CN1113540C	2003-07-02	1997-06-10	Video decoder having interfacing function for picture synchronization

What is claimed and desired to be secured by United States Letters Patent is:

1. A video decoder having an interfacing function for picture synchronization, the video decoder comprising:

- packet separation means for separating packetized elementary stream (PES) packet data of a desired program from a received transport stream (TS), and outputting the PES packet data;
- interfacing means for receiving the PES packet data output from the packet separation means, and outputting time stamp information and encoded video data contained in the PES packet data; and
- decoding means for receiving the data output from the interfacing means, decoding the encoded video data using the time stamp information and displaying the decoded video data;
- wherein said interfacing means multiplexes the extracted time stamp information with a start code of a corresponding picture and outputs the encoded video data in units a size of a picture and in a bit stream form.

2. The video decoder according to claim 1, further comprising storage means coupled between said interfacing means and said decoding means, for receiving the output data of said interfacing means and storing the received data.

3. A video decoder having an interfacing function for picture synchronization, the video decoder, comprising:

- packet separation means for separating packetized elementary stream (PES) packet data of a desired program from a received transport stream (TS), and outputting the PES packet data;
- interfacing means for receiving the PES packet data output from the packet separation means, and outputting time stamp information and encoded video data contained in the PES packet data; and
- decoding means for receiving the data output from the interfacing means, decoding the encoded video data using the time stamp information and displaying the decoded video data;
- wherein said interfacing means comprises:
    - o an input buffer for storing the PES packet data received from said packet separation means;
    - o a PES filter for extracting the encoded video data from the PES packet data stored in said input buffer; and
    - o multiplexing means for extracting the time stamp information contained in the PES packet data stored in said input buffer, multiplexing the extracted time stamp information with the corresponding encoded video data extracted by the PES filter, and outputting the multiplexed result to said decoding means.

4. The video decoder according to claim 3, further comprising storage means coupled between said interfacing means and said decoding means, for receiving the output data of said interfacing means and storing the received data.

5. The video decoder according to claim 3, wherein said multiplexing means comprises:

- a time stamp extractor for extracting the time stamp information contained in the PES packet data stored in said input buffer;
- a picture start code (PSC) detector for detecting a picture start code PSC of the encoded video data contained in the PES packet data stored in said input buffer; and
- a multiplexer for multiplexing the time stamp information extracted from said time stamp extractor with the corresponding picture start code of the encoded video data output from said PES filter, based on the picture start code PSC detected in the PSC detector, and outputting the multiplexed result.

6. The video decoder according to claim 5, wherein said multiplexer multiplexes the picture start code and the time stamp information to be adjacent to each other.

7. The video decoder according to claim 6, wherein said multiplexer multiplexes the time stamp information with a corresponding identification code following an uppermost 24 bits of the PSC.

8. The video decoder according to claim 7, wherein said time stamp information includes a presentation time stamp (PTS) and a decoding time stamp (DTS).

9. The video decoder according to claim 8, wherein said decoding means uses the DTS information obtained by decoding the output data of said multiplexer, to start a decoding operation of the encoded video data in units of a corresponding picture and uses the PTS information to start displaying the decoded video data in a unit of a picture.

10. A video decoder having an interfacing function for picture synchronization, the video decoder comprising:

- a packet separator operative to separate packetized elementary stream (PES) packet data of a desired program from a received transport stream (TS), and output the PES packet data;
- an interface operative to receive the PES packet data output from the packet separator, and output time stamp information and encoded video data contained in the PES packet data; and
- a decoder operative to receive the data output from the interface and decode the encoded video data using the time stamp information;
- wherein said interface is further operative to multiplex the extracted time stamp information with a start code of a corresponding picture and output the encoded video data in units a size of a picture and in a bit stream form.

11. The video decoder according to claim 10, further comprising a memory coupled between said interface and said decoder, operative to receive the output data of said interface and store the received data.

12. A video decoder having an interfacing function for picture synchronization, the video decoder comprising:

- a packet separator operative to separate packetized elementary stream (PES) packet data of a desired program from a received transport stream (TS), and output the PES packet data;
- an interface operative to receive the PES packet data output from the packet separator, and output time stamp information and encoded video data contained in the PES packet data; and
- a decoder operative to receive the data output from the interface and decode the encoded video data using the time stamp information;
- wherein said interface comprises:
    - an input buffer for storing the PES packet data received from said packet separator;
    - a PES filter for extracting the encoded video data from the PES packet data stored in said input buffer; and
    - a multiplexer operative to extract the time stamp information contained in the PES packet data stored in said input buffer, multiplex the extracted time stamp information with the corresponding encoded video data extracted by the PES filter, and output the multiplexed result to said decoder.

13. The video decoder according to claim 12, wherein said multiplexer comprises:

- a time stamp extractor operative to extract the time stamp information contained in the PES packet data stored in said input buffer;
- a picture start code (PSC) detector for detecting a picture start code PSC of the encoded video data contained in the PES packet data stored in said input buffer; and
- a multiplexer operative to multiplex the time stamp information extracted from said time stamp extractor with the corresponding picture start code of the encoded video data output from said

PES filter, based on the picture start code PSC detected in the PSC detector, and output the multiplexed result.

14. The video decoder according to claim 13, wherein said multiplexer multiplexes the picture start code and the time stamp information to be adjacent to each other.

15. The video decoder according to claim 14, wherein said multiplexer multiplexes the time stamp information with a corresponding identification code following an uppermost 24 bits of the PSC.

16. The video decoder according to claim 15, wherein said time stamp information includes a presentation time stamp (PTS) and a decoding time stamp (DTS).

17. The video decoder according to claim 16, wherein said decoder uses the DTS information, obtained by decoding the output data of said multiplexer, to start a decoding operation of the encoded video data in units of a corresponding picture and uses the PTS information to start displaying the decoded video data in a unit of a picture.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a video decoder for restoring video data of an encoded and multiplexed bit stream, and more particularly, to a video decoder having an interfacing function for picture synchronization which can synchronize a decoding and display operation in units of a picture to precisely control the video decoder. The present application is based on Korean Patent Application No. 96-20792, which is incorporated herein by reference.

2. Description of the Related Art

The MPEG-II standard for motion picture compression presents the standard for a high-definition television. A video decoder adopting the standard decodes and displays video data among transport stream (TS) data obtained by multiplexing video data and audio data encoded in the form of each bit stream with respect to a plurality of programs. The TS stream includes video data encoded in units the size of a picture, a picture start code (PSC) indicating a picture start, and time stamp information, in the form of a packetized elementary stream (PES) packet. The PES is defined in the system standard DIS 13818-1 in the MPEG-II standard in order to packetize and multiplex video data encoded in the bit stream form in units of a constant length or variable length packet. The time stamp information relates to the decoding time and reproducing output time, which includes decoding time stamp (DTS) and presentation time stamp (PTS) information, respectively. The PTS and DTS information is contained in the PES packet header, and may not be transferred for every picture. When a transfer is made, the PTS and DTS information is transferred with respect to an intra-coded I-picture and a predictive-coded P-picture. Only the PTS is transferred with respect to a bidirectionally predictive-coded B-picture since the PTS and the DTS are the same. The video decoder decodes the PTS and DTS information to thereby perform a control operation in units the size of a picture. The PTS represents the time for displaying the decoded picture and the DTS represents the start time for decoding a bit stream. Thus, a normal operation of the video decoder can assuredly prevent decoding and display operations from temporally leading and lagging using the DTS and PTS whenever the picture is decoded and displayed.

FIG. 1 is a block diagram showing a general video decoder. The general video decoder receives the TS stream transmitted via a channel. A TS demultiplexer 11 separates the video data and the time stamp information encoded in the bit stream form, corresponding to a program selected by a user, from the received TS stream and outputs the separated result. A bit buffer 12 stores the video data encoded in the bit stream form which has been separated by the TS demultiplexer 11. A time stamp buffer 13 stores the time stamp information which has been separated by the TS demultiplexer 11. The bit buffer 12 outputs the stored video data encoded in the bit stream form whenever a variable-length decoder 14 requests data. At the same time, the time stamp buffer 13 outputs the time stamp information corresponding to the video data output from the bit buffer 12. Accordingly, the timing of the decoding and display operations of the output video data is synchronized in units the size of a picture.

The variable-length decoder 14 decodes the time stamp information output from the time stamp buffer 13 to obtain the PTS and DTS information, outputs the PTS information to a display (not shown), and performs a variable-length-decoding operation with respect to the encoded video data output from the bit buffer 12 using the DTS information. An inverse quantizer 15 and an inverse discrete cosine transformer (IDCT) 16 inversely quantizes and inversely discrete cosine transforms the variable-length-decoded video data. The data output from the IDCT 16 is supplied to an adder 17. A frame memory 18 stores the video data output from the adder 17. A motion compensator 19 performs a motion compensation operation with respect to the video data stored in the frame memory 19 using a transmitted motion vector (not shown). The adder 17 adds the inversely quantized and inverse discrete cosine transformed data and the video data motion-compensated in the motion compensator 19. The video data output from the adder 17 is restored video data, and is both output to the display and stored in the frame memory 18. The display starts a display operation of the

corresponding restored video data using the PTS information obtained from the variable-length decoder 14.

However, the above video decoder stores the bit stream video data and the time stamp information in different buffers and reads the same therefrom. Thus, when the time stamp information is used for decoding and display operation control of the bit stream video data of the corresponding picture, synchronization may be lost. Furthermore, it is difficult to effectively use the time stamp information for decoding and display operation control when errors occur, which causes difficulty in reproducing a normal picture.

SUMMARY OF THE INVENTION

To solve the above problems, it is an object of the present invention to provide a video decoder having an interfacing function for picture synchronization which multiplexes time stamp information with a corresponding picture start code and store the multiplexed result together with the encoded video data of the picture in a bit buffer to thereby maintain synchronization with the time stamp information for every picture, without separately processing the encoded video data and the time stamp information.

To accomplish the above object of the present invention, there is provided a video decoder having an interfacing function for picture synchronization, the video decoder comprising: packet separation means for separating packetized elementary stream (PES) packet data of a desired program from a received transport stream (TS); interfacing means for receiving the PES packet data output from the packet separation means, and outputting time stamp information and encoded video data contained in the PES packet data; and decoding means for receiving the data output from the interfacing means, decoding the encoded video data using the time stamp information and displaying the decoded video data via a display.

Brief Description of the Drawings

In order that the manner in which the above-recited and other advantages and objects of the invention are obtained, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

FIG. 1 is a block diagram of an exemplary system for implementing the invention that includes a general purpose computing device in the form of a conventional personal computer.

FIGS. 2A and 2B show two different configurations of an IP network. FIG. 2A shows a stand-alone IP network isolated from any other networks, while FIG. 2B shows the isolated IP network of FIG. 2A that has access to other networks through one of the IP hosts acting as a gateway.

FIGS. 3A-3C illustrate the three different configurations of an IP address wherein a certain number of bits in the fixed 32-bit IP address space are assigned to a network portion, while other bits are assigned to a host portion.

FIG. 4 is a flow chart depicting the processing steps that a IP host would take to automatically generate an IP address for use on an IP network where an IP address server is initially not present.

FIGS. 5A-5C illustrate the use of IP hosts that automatically generate their IP addresses while on a stand-alone IP network and then subsequently receive an assigned IP address when they are connected to a corporate IP network. FIG. 5A illustrates the IP network address automatically generated in each case. FIG. 5B shows the connection with the corporate IP network and Internet, including the addition of the assigned IP addresses from an IP address server available on the corporate network, and FIG. 5C shows the discontinued use of the automatically generated IP addresses.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

As used herein, the term "hashing function" refers to those functions known in the art that systematically covert one multi-bit representation into another, usually smaller, multi-bit representation. A hashing function is said to be deterministic if it generates the same results from the same input.

As used herein, the term "IP address server" refers to a host on the IP network that will manage and

assign IP addresses to other hosts. Such a server may use the Dynamic Host Configuration Protocol (DHCP) to assign IP addresses to the network hosts.

As used herein, the term "host" refers to any separately cognizable entity on an IP network. While there is typically a single IP address for each host, it is conceivable that a single host may have multiple IP addresses. References throughout the application to information available to the host refers to information that may be accessed or generated by the host processor and includes by way of example and not limitation, the network interface card IEEE 802 Ethernet address on conventional PCs and other Ethernet compatible devices, random number generators available from the host operating system, hardware serial numbers available on the host, etc.

FIG. 1 and the following discussion are intended to provide a brief, general description of a suitable computing environment in which the invention may be implemented. Although not required, the invention will be described in the general context of computer-executable instructions, such as program modules, being executed by a personal computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

With reference to FIG. 1, an exemplary system for implementing the invention includes a general purpose computing device in the form of a conventional personal computer 20, including a processing unit 21, a system memory 22, and a system bus 23 that couples various system components including the system memory to the processing unit 21. The system bus 23 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. The system memory includes read only memory (ROM) 24 and random access memory (RAM) 25. A basic input/output system 26 (BIOS), containing the basic routines that helps to transfer information between elements within the personal computer 20, such as during start-up, may be stored in ROM 24. The personal computer 20 may also include a hard disk drive 27 for reading from and writing to a hard disk, not shown, a magnetic disk drive 28 for reading from or writing to a removable magnetic disk 29, and an optical disk drive 30 for reading from or writing to removable optical disk 31 such as a CD ROM or other optical media. The hard disk drive 27, magnetic disk drive 28, and optical disk drive 30 are connected to the system bus 23 by a hard disk drive interface 32, a magnetic disk drive-interface 33, and an optical drive interface 34, respectively. The drives and their associated computer-readable media provide nonvolatile storage of computer readable instructions, data structures, program modules and other data for the personal computer 20. Although the exemplary environment described herein employs a hard disk, a removable magnetic disk 29 and a removable optical disk 31, it should be appreciated by those skilled in the art that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, random access memories (RAMs), read only memories (ROM), and the like, may also be used in the exemplary operating environment.

A number of program modules may be stored on the hard disk, magnetic disk 29, optical disk 31, ROM 24 or RAM 25, including an operating system 35, one or more application programs 36, other program modules 37, and program data 38. A user may enter commands and information into the personal computer 20 through input devices such as a keyboard 40 and pointing device 42. Other input devices (not shown) may include a microphone, joy stick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 21 through a serial port interface 46 that is coupled to the system bus, but may be connected by other interfaces, such as a parallel port, game port or a universal serial bus (USB). A monitor 47 or other type of display device is also connected to the system bus 23 via an interface, such as a video adapter 48. In addition to the monitor, personal computers typically include other peripheral output devices (not shown), such as speakers and printers.

The personal computer 20 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 49. The remote computer 49 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the personal computer 20, although only a memory storage device 50 has been illustrated in FIG. 1. The logical connections depicted in FIG. 1 include a local area network (LAN) 51 and a wide area network (WAN) 52 that are presented here by way of example and not limitation. Such networking environments are commonplace in offices enterprise-wide computer networks, intranets and the Internet.

When used in a LAN networking environment, the personal computer 20 is connected to the local network 51 through a network interface or adapter 53. When used in a WAN networking environment,

the personal computer 20 typically includes a modem 54 or other means for establishing communications over the wide area network 52, such as the Internet. The modem 54, which may be internal or external, is connected to the system bus 23 via the serial port interface 46. In a networked environment, program modules depicted relative to the personal computer 20, or portions thereof, may be stored in the remote memory storage device. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

Referring to FIG. 2A, a number of IP hosts 60a-60e are shown interconnected in a stand-alone IP network. Each of the IP hosts 60a-60e will have a separate IP address and be able to communicate with each of the other IP hosts. Furthermore, each of the IP hosts may be a conventional computer system as described in connection with FIG. 1 running appropriate IP communication software. The IP communications software and the network interface 53 together constitute a means for communicating over an IP network. Note that the network shown in FIG. 2A could be a LAN in a small department of a company, in a small business, or in a home.

Referring to FIG. 2B, the stand-alone network shown in FIG. 2A is interconnected with another network through the IP host 60c. The existing network(s) may have an IP address server that could assign addresses to the IP hosts 60a-60e. A situation as shown in FIG. 2B may occur when a department LAN is interconnected to the corporate network. In this a situation, each of the IP hosts 60a-60e would now receive an IP address from the IP address server on the existing network 62 and may eventually cease using the automatically generated IP address; such a scenario will be shown hereafter in more detail in connection with FIGS. 5A-5C.

Referring now to FIGS. 3A-3C, three different formats of an IP address are shown. An IP address is divided into three portions or regions: a format indication portion, a network identifying portion, and a host identifying portion. The relative number of bits assigned to each of these portions will determine the maximum number of entities that can be supported by a particular addressing format. For example, allocating more bits to the host identifying portion will increase the number of hosts that can exist on a single network. Reducing this number of bits will allow more networks to be supported but reduce the maximum number of hosts per network. The three standard formats provides flexibility and all IP hosts are able to recognize each of the three formats.

FIG. 3A shows an IP address format having a relatively small network identifying portion 64 of 7 bits and a correspondingly larger host identifying portion 66 having 24 bits. This particular IP address format is indicated by reserved bit 68 having a value of zero.

In like manner, FIG. 3B shows an IP address format wherein the network identifying portion 70 is composed of 14 bits and the host identifying portion 72 is composed of 16 bits and is indicated by a value of `10` for the reserved bits 74. The IP address format of FIG. 3B gives approximately equal addressing capability to designate the network and the host.

Finally, FIG. 3C shows an IP address format wherein the network identifying portion 76 has 21 bits, the host identifying portion 78 has 8 bits, and the indicating bits 80 have a value of `110`. Naturally, the format of FIG. 3C will give greater addressing capability to the network identifying portion at the expense of being able to identify a smaller number of hosts. In each of the three IP addressing formats shown in FIGS. 3A-3C, there is a network identifying portion that indicates the network whereon the host resides and a host identifying portion that identifies the particular host on the designated IP network.

Any of the three formats may be used when automatically generating an IP address since all IP hosts recognize each format and different IP hosts may use different formats on any given IP network. It may be advantageous in some circumstances to select the format shown in FIG. 3A since it will allow the most possible hosts to exist on a single network

For an automatically generated IP address, some convention needs to be chosen so that the network portion is the same for all IP hosts on the same network. For example, each IP host 60a-60e shown in FIG. 2A that automatically generates its IP address would have the same number identified in the network portion regardless of format in order to conform with the IP address standard.

One embodiment of the invention will always select the value of "10" for the network identifying portion of the automatically generated IP address. This value, known as "net 10," was the value used for the DARPA net, the historical predecessor to the current day Internet. In order to reduce confusion, the IANA will not assign "net 10" to any existing organization. Therefore, the "net 10" value for the network identifying portion of an IP address is relatively safe from conflict and has been used for debugging purposes and for internal networks within a corporate network structure.

Another embodiment will use a specifically reserved value for the network identifying portion of the IP address that has been assigned by the IANA for automatically generated IP network addresses. One such reserved or assigned value is 169.254.

Yet another way of consistently determining or obtaining the network identifying portion of an IP address is to use a network protocol for obtaining the information for the portion directly or indirectly from other devices attached to the network. Those skilled in the art will also appreciate that other mechanisms may also exist that would allow consistent and predictable generation of the same

network identifying portion of an IP address that would work with the present invention are possible.

Referring now to FIG. 4, a flow chart showing the process steps taken by an actual IP host according to one embodiment of the present invention is shown. IP address acquisition begins at step 82 and it is first determined whether an IP address server is present at step 84. If there is an IP address server, there is no need to automatically generate an IP address and subsequently an IP address will be received from the IP address server at step 86.

To test for the IP address server at step 84, the IP host will send messages over the IP network according to a certain protocol, such as DHCP, to identify itself to the IP address server. The IP address server may then participate in a session that allows the assignment of an IP address from the IP address server to the IP host, again according to a prescribed protocol, such as DHCP. Using such a protocol is a means for determining the presence or absence or an IP address server. After receiving the IP address at step 86, the host will use that IP address indefinitely and the IP address acquisition operation terminates at step 88.

If an IP address server is not available at step 84 as the host boots up on the IP network and gives the appropriate protocol to signal a desire for an IP address, a determination is made at step 90 whether the host is configured for automatic IP address generation. If the host is not configured for automatic IP address generation, an error condition will be signaled at step 91 before ending IP address acquisition at step 88. The error condition will signal the user that IP network communications are not possible at that particular time.

After determining that the host is configured for automatic IP address generating at step 90, the network identifying portion of the IP address will be selected at step 92. As explained previously, one embodiment chooses the "net 10" value for the network identifying portion of the IP address. An important element is to consistently select the same network number value regardless of the method implemented for making that selection. For example, a fixed number may be always chosen (e.g., "net 10" or a "reserved" value), a common number may be accessed in a predetermined manner from a file, from the network, or from some other source, etc., and all such methods would be considered means for consistently selecting the network identifying portion of an IP address. Furthermore, as explained previously, any format of the IP address as shown in FIGS. 3A-3C may be used.

Next, the host identifying portion of the IP address is deterministically generated at step 94. One preferred way of generating the host identifying portion of the IP address is to begin with a known value available to the IP host and use a deterministic hashing algorithm to generate the host portion of the IP address. For example, the network interface card IEEE 802 Ethernet address, which is too large to fit directly into the host identifying portion of an IP address, may be deterministically hashed to arrive at a value that will fit within the length confines of the host portion of the IP address. Other values available to an IP host may also be used, such as random number generation from the host operating system, mother board serial numbers, etc. The IEEE 802 Ethernet address value is preferred due to its globally unique nature that, with the use of an appropriate deterministic hashing algorithm, would result in less usage conflicts for an automatically generated IP address and its high availability in PC networks.

Any other method that results in a high likelihood that each host will generate a unique, deterministic value for the host identifying portion may be used and would in like manner be considered a means for deterministically generating the host identifying portion of an IP address. It is preferred that the process be computationally efficient as well to eliminate large delays in the generation process.

A preferred hashing algorithm will be deterministic in the sense that a given input will generate the same output each time the algorithm is applied. The hashing algorithm should also be capable of being rehashed in the event that a usage conflict occurs wherein the host identifying portion generated by the original hash corresponds to an address already in use. Rehashing is generally performed by feeding the results of the first hash back into the algorithm to generate a new output. Finally, the hashing algorithm should have a good avalanche characteristic in order to more evenly spread out the hash results across the available values.

Before using the automatically generated IP address, it is tested for conflicting usage by other IP hosts on the network at step 96. One way of or means for doing such testing is for the IP host to send a message out on the IP network addressed to the automatically generated IP address. If a response is received, then another host on the IP network is using that particular IP address. Those skilled in the art will recognize that other ways and variations for determining IP address conflicts exist, any one of which may be used for purposes of testing an IP address for conflicting usage. All that is required is for the host to determine if another host on the network is already using the generated IP address.

If an IP address usage conflict is identified at step 98, the IP address is regenerated in the manner explained previously in connection with steps 92 and 94. Note that since "net 10" is chosen for the network identifying portion of the IP address for this embodiment, only the host identifying portion of the IP address need be regenerated. This is typically done by rehashing the IEEE 802 Ethernet address and testing the new rendition of the generated IP address having the newly generated host identifying portion for usage conflict again at step 96. This process will occur for a predetermined

number of times or until no conflict usage is identified at step 98. Note that rehashing is only one means of regenerating the IP address and those skilled in the art will recognize others that accomplish the same purpose.

At step 100, the generated IP address is used by the IP host for communicating on the IP network. Also at step 100 periodic tests are made to determine if an IP address server has become available over the IP network since the IP address was generated or to determine if there is conflicting usage of the IP address. Each of these two situations may indicate a change for the IP host in using the generated IP address depending on actual implementation. One way or means for monitoring conflicting IP address usage is to periodically send a message to the generated IP address and wait for a response as explained previously in connection with step 96.

If a conflict usage is identified at step 102 because the generated IP address is also being used by another IP host, then an error indication is signaled at step 104 before ending at step 88. Because this situation may be extremely detrimental and because the situation is very uncommon due to previous conflict testing at step 96, one embodiment will simply shut down the IP stack and no longer process network messages. The host user is then made aware through the user interface that a catastrophic error has occurred. Alternatively, rather than shut down the IP stack with an error, other embodiments may attempt to generate a new IP address.

If no conflicting usage is identified at step 102 and no IP address server has become available at step 106, use of the generated IP address continues at step 100 until more periodic tests are made. In one embodiment, testing for an IP address server is done by sending messages according to the DHCP protocol. Note that this test can be the same test described in conjunction with step 84 above and serves as a means for ascertaining if an IP address server later becomes available.

If an IP address server later becomes available at step 106, the IP host will interact with the address server in order to get another IP address at step 108. DHCP is one protocol that may be implemented as a means for requesting and receiving an IP address for an IP host. This address will then be used by the IP host while use of the generated IP address will eventually cease at step 110 before ending address acquisition at step 88. Those skilled in the art will see that variations may exist wherein multiple IP addresses are used by a single IP host that will allow the automatically generated IP address to co-exist with the IP address assigned by the IP address server.

When ceasing to use the generated IP address at step 110, implementations may vary as to how quickly this may be achieved. For example, use of the generated IP address may be immediately cut off upon receipt of the IP address from the IP address server; only existing sessions with the generated IP address may be maintained thereby gradually decreasing the use of the generated IP address; or, finally, both addresses may be used simultaneously. Software running on the IP host will control the continued use or disuse (whether gradual or immediate) of the generated IP address according to a variety of factors. This software constitutes a means for gradually discontinuing the use of the generated IP address when gradual disuse is mandated by the implementation and the circumstances.

Referring now to FIGS. 5A-5C, an example implementation of the present invention is shown as IP addresses are initially generated on a local area network that is eventually integrated into the corporate-wide network having an IP address server. Furthermore, the corporate network is connected to the Internet. FIG. 5A shows the state of the independent local area network after each of the three IP hosts has automatically generated their respective IP addresses. FIG. 5B then shows the state of the local area network after integration into the corporate IP network and wherein each of the three IP hosts has received an address from the corporate IP address server and showing the simultaneous use of the generated IP address and the assigned IP address as the generated IP address is gradually phased out. Finally, FIG. 5C shows the state of the local area network after the generated IP addresses are no longer in use for each of the three nodes.

Referring to the network 112 of FIG. 5A, each IP host has automatically generated and tested an IP address according to the processing steps explained previously in connection with the flowchart of FIG. 4. Specifically, host 114 is using generated IP address 116 having a value of 10.0.0.107, host 116 is using generated IP address 120 having a value of 10.0.3.255, and host 122 is using generated IP address 124 having a value of 10.0.0.7.

One or more of the generated IP addresses may have had an initial conflict but such has been resolved by rehashing to get a second or more order hash of the IEEE 802 Ethernet address found in the network interface card of the conventional PC that makes up each respective host. In most instances, each time a host is powered on and "active" on the network 112, it will have the same IP address due to the deterministic nature of the hashing algorithm generating the host identifying portion of the address. This configuration allows the respective IP hosts to be physically attached to the network hardware, powered on, and immediately communicating on the network 112 without the need of any explicit configuration with respect to an IP address.

Referring to FIG. 5B, the network 112 is integrated into a corporate IP network 126 through suitable router hardware 128. Note that, alternatively, one of the IP hosts, such as host 120 could function as a router to make the logical integration of the network 112 into the corporate IP network 126 as

represented by dashed line 130. Furthermore, the corporate IP network 126 is part of the world-wide Internet.

Once connected onto the corporate IP network, an IP address server will become available to each respective host 114, 118, and 122. Each host will "discover" the IP address server during execution of a periodic test as described in connection with step 100 of FIG. 4. After going through the appropriate interaction, such as DHCP, each respective host 114, 118, and 122 will receive an assigned IP address from the IP address server. Specifically, host 114 is using assigned IP address 132 having a value of 81.0.0.1, host 116 is using assigned IP address 134 having a value of 81.0.0.2, and host 122 is using assigned IP address 136 having a value of 81.0.0.3.

The assigned IP address will now be used while the generated IP address will be gradually phased out from use. In one embodiment, existing sessions using the IP address will continue until ended while all new sessions will use the assigned IP address received from the IP address server. With respect to the configuration shown in FIGS. 5A-5C, it is only the hosts within the network 112 that will be using the generated IP addresses. Note also that once the network 112 is attached to the corporate IP network 126, a chance of conflicts occurs with existing addresses on the corporate IP network 126 that could lead to the error situation discussed with processing steps 102 and 104 of FIG. 4.

Referring to FIG. 5C, each respective node 114, 118, and 122 has ceased using the generated IP address and is only using the assigned IP address. Each time one of the hosts 114, 118, or 122 powers up and becomes active on the network 112 (and subsequently the corporate IP network 126) it will use its respective assigned IP address.

The present invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrated and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

FIG. 1 (PRIOR ART)

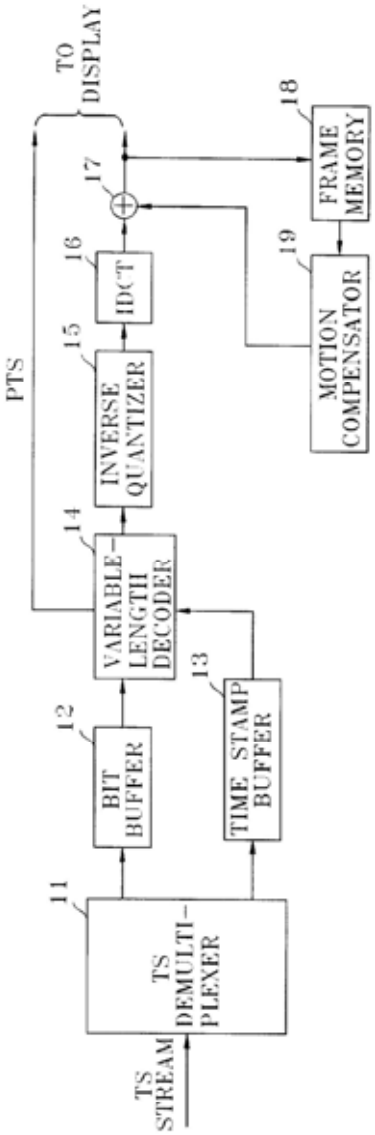
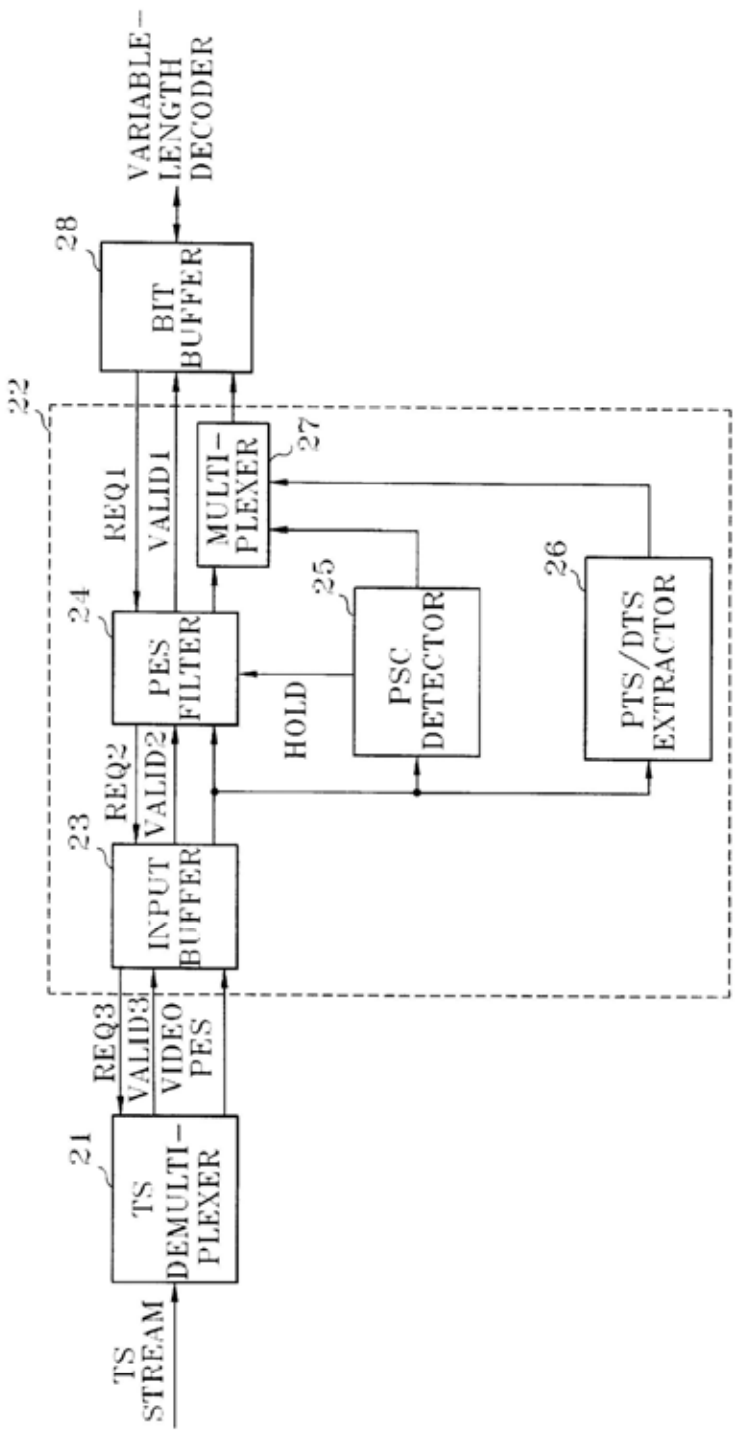


FIG. 2


TS STREAM
21
TS DEMULTI-PLEXER
REQ3
VALID3
VIDEO PES
23
INPUT BUFFER
REQ2
VALID2
24
PES FILTER
HOLD
25
PSC DETECTOR
26
PTS/DTS EXTRACTOR
REQ1
VALID1
27
MULTI-PLEXER
22
28
BIT BUFFER
VARIABLE-LENGTH DECODER

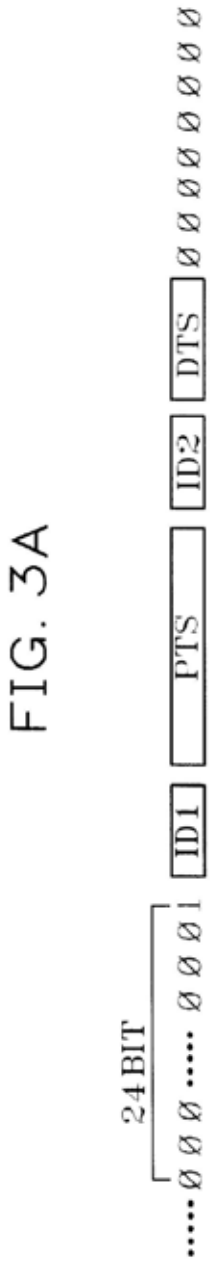


FIG. 3A

FIG. 3B

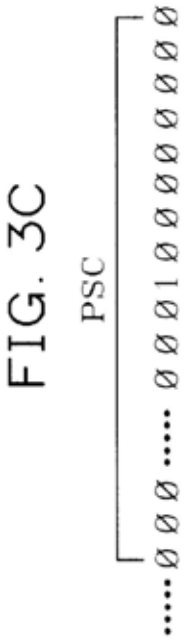


FIG. 3C

Patent	Pub.Date	Inventor	Assignee	Title
US5396497	1995-03	Veltman	Sony Corporation	Synchronization of audio/video information
US5517250	1996-05	Hoogenboom et al.	General Instrument Corporation of Delaware	Acquisition of desired data from a packetized data stream and

				synchronization thereto
US5521927	1996-05	Kim et al.	Electronics and Telecommunications Research Institute	Elementary stream packetizing unit for MPEG-2 system
US5598415	1997-01	Nuber et al.	General Instrument Corporation of Delaware	Transmission of high rate isochronous data in MPEG-2 data streams
US5726989	1998-03	Dokic	Stellar One Corporation	Method for ensuring synchronization of MPEG-1 data carried in an MPEG-2 transport stream
US5801781	1998-09	Hiroshima et al.	Fujitsu Limited	Apparatus for converting moving picture stream of MPEG1 to transport stream of MPEG2
US5812976	1998-09	Ryan	Matsushita Electric Corporation of America	System and method for interfacing a transport decoder to a bitrate-constrained audio recorder
US5832256	1998-11	Kim	Samsung Electronics Co., LTD.	System clock recovery apparatus for MPEG decoding system
US5886736	1999-03	Chen	General Instrument Corporation	Synchronization of a stereoscopic video sequence
US5898695	1999-04	Fujii et al.	Hitachi, Ltd.	Decoder for compressed and multiplexed video and audio data
US5960006	1999-09	Maturi et al.	LSI Logic Corporation	MPEG decoding system adjusting the presentation in a predetermined manner based on the actual and requested decoding time