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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
24 October 2002 (24.10.2002)

PCT

(10) International Publication Number
WO 02/084933 A2

(51) International Patent Classification[7]: H04L 5/00

(21) International Application Number: PCT/NL02/00253

(22) International Filing Date: 18 April 2002 (18.04.2002)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
1017870 18 April 2001 (18.04.2001) NL

(71) Applicant *(for all designated States except US)*: **NONEND INVENTIONS N.V.** [NL/NL]; Van Engelenweg 23, Curaçao (AN).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: **VAN OLDENBORGH, Marc** [NL/NL]; Omval 263, NL-1096 HR Amsterdam (NL). **GNIRREP, Martijn** [NL/NL]; Bernard Shawsingel 282, NL-1102 VE Amsterdam (NL).

(74) Agent: **DE HOOP, Eric**; Octrooibureau Vriesendorp & Gaade, P.O. Box 266, NL-2501 AW The Hague (NL).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ (utility model), CZ, DE (utility model), DE, DK (utility model), DK, DM, DZ, EC, EE (utility model), EE, ES, FI (utility model), 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, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK (utility model), SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZM, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, 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, GQ, GW, ML, MR, NE, SN, TD, TG).

Declaration under Rule 4.17:
— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii)) for the following designations 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, OM, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TN, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZM, ZW, ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, 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, GQ, GW, ML, MR, NE, SN, TD, TG)*

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

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.

WO 02/084933 A2

(54) Title: METHOD FOR INVERSE MULTIPLEXING

(57) Abstract: The invention relates to a method for multiplexing digital data, wherein a packet of digital data is simultaneously sent from front to rear and from rear to front. Such a method can be used in cordless telephones, computer networks, such as among others the internet, and other networks of devices wherein digital data are exchanged.

Method for inverse multiplexing

Field of the invention

The invention relates to a method for multiplexing digital data, and software for that purpose.

Background

In practice very many methods to send data are known, particularly through a network. An example are digital data that are sent through the internet, but also digital telephone signals, such as for instance in case of cordless GSM telephones.

In practice there are also various methods known for multiplexing digital data.

The data are sent via a medium in the form of data packets. This can take place via a physical cabling in for instance copper or glass fibre, or via infrared or radio waves.

Because the capacity, often coupled to bandwidth, of the media over which the data are being sent mostly is too small, the data are often compressed. However, this has often appeared not to be sufficient.

Additionally many of these methods are aimed at optimising the data transfer between two computers.

Summary of the invention

It is among others an object of the present invention to at least partially solve these problems. To that end the invention provides a method for multiplexing digital data, wherein a packet of digital data is simultaneously sent from front to rear and from rear to front. Additionally the invention provides a method for multiplexing digital data, wherein a packet of digital data is simultaneously sent from the beginning of the packet towards the end and from the end towards the beginning. Additionally the invention provides a method for multiplexing digital data, wherein simultaneously a packet of digital data is sent and the same packet is sent backwards.

By splitting the data stream in two simultaneous streams, wherein the one stream starts sending the data from the front and works its way to the rear, and the other stream works its way from the rear to the front, the possibility is given to very quickly send data between and to for instance computers. In most of the cases, the invention relates to sending this data via electromagnetic waves, eg electronically or optically.

Another advantage of the method is that no other control signals or techniques are necessary to reconstruct the entire signal or data packet or to adjust both streams to each other: the signal or data packet is complete when the two streams meet each other, or when the data buffer is full. Delays in one of either lines do not lead to loss of signal.

In many cases the data connection is a-symmetrical: the transmission capacity is smaller than the receiving capacity. In the method according to the invention it is possible, despite the smaller transmission capacity, to nonetheless use the full receiving capacity. This may for instance be of importance in so-called streaming broadcasts through the internet, wherein through the internet digital radio broadcasts and in future even television or video can take place. Both data streams can also enter via various lines.

For instance the one data stream can for example enter via a telephone line and the second data stream via a cable, the electricity grit or cordless via GSM. It is also possible to let the data streams enter via one cable by means of physical multiplexing. The invention therefore actually offers a specific form of digital multiplexing.

Preferably the present invention relates to a method as described, wherein a first device sends the data from front to rear to a third device, and a second device sends the same data from rear to front to the third device. As a result the third device is able to have all data available very quickly.

Preferably the third device places the data in a data buffer the size of the packet, and sends a signal to the first and second device when either the buffer is full, or stops sending confirmations until the buffer is full. In this way the coordination between both streams is very simple.

It is preferred that in a method according to the invention a first device sends data from front to rear to a second device, and simultaneously backwards to a third device. As a result the possibility is offered to very quickly provide two devices with all data, with an optimally used bandwidth. In this method it is preferred that the second device and third device immediately at receipt forward the data they received from the first device to each other. As a result both devices can optimally use their bandwidth and transmission capacity.

In said method it is preferred that the second and third device have been provided with a data buffer the size of the packet, wherein the received data are placed in the data buffer and the first and second device send a signal to the first device when the respective data buffer is full.

Additionally the invention relates to a method for sending a data packet to a first device in an organic (also called 'ad-hoc') data network of devices,

wherein the devices have been provided with a data processing unit, a data buffer and software having receiving routines for receiving data packets from at least two transmitting devices in the data network, wherein at least two other devices in the network simultaneously send complementary data packets to the first device which added together form the data packet.

Preferably said software has further been provided with transmission routines for transmitting data packets, received from the transmitting device or devices in the data network to at least one receiving device that is connected to the data network, independent of the transmitting device or devices.

Additionally the invention relates to a method for receiving digital data, wherein a device provided with data storage means creates a data buffer in the data storage means the size of a packet of digital data, and simultaneously receives a first stream of digital data and receives a second stream of digital data, wherein the device fills the data buffer from front to rear with the first stream of digital data and fills the data buffer from rear to front with the second stream of digital data.

Preferably the device informs the source or sources of the streams of digital data when a data buffer is full. As a result the coordination is simple.

Additionally the invention relates to a method for sending digital data, wherein a device provided with data storage means creates a data buffer in the data storage means, stores digital data in the data buffer, and from the front of the data buffer and the rear of the data buffer sends the digital data in two streams.

Preferably the device stops sending after receipt of a signal. As a result the coordination is again simple.

Additionally the invention relates to software provided with routines for carrying out the method according to one of methods mentioned above.

From the above description, in combination with the figures and their description, it will immediately be clear to an expert which routines are necessary to that end, and how said routines have to function with respect to each other. Such software may of course be immediately implemented in hardware, for instance in a PROM, EPROM or the like.

Additionally, the invention relates to an software for sending a packet of digital data, comprising a first transmission routine for sending a first stream of digital data starting from the front of the packet of digital data and a second transmission routine for sending a second stream of digital data starting from the end of the packet of digital data

Furthermore, the invention relates to software for receiving a packet of digital data, comprising a first receiving routine for receiving a first stream of digital data and a second recieving routine for simultaneously receiving a second stream of digital data, and a first storing routine for storing the first stream of digital data in a memory starting at the front of the memory and filling the memory towards the end, and a second storing routine for storing the second stream of digital data starting at the end of the memory and filling the memory towards the front, and a stop routine for ending the receiving of digital data when the memory is full.

Furthermore, the invention relates to an apparatus for sending a packet of digital data, comprising memory means for storing the packet of digital data, first sending means for sending a first stream of digital data, starting at the front of the memory means and second sending means for sending a second stream of digital data, starting at the end of the memory means.

Furthermore, the invention relates to an apparatus for receiving a packet of

digital data, comprising memory means for storing the packet of digital data, first receiving means for receiving a first stream of digital data, and storing it in said memory means, starting from the front of the memory means, and second receiving means for receiving a second stream of digital data, and storing it in said memory means, starting from the back of the memory means.

Additionally the invention relates to a carrier provided with software as described, and to a device provided with software as described.

In a specific embodiment of the invention, packets of digital data are sent in the form of bitstreams. In another embodiment, the packets are divided into smaller sub-packets, for instance numbered 1..n. In this case, the sub-packets are sent 1, 2, etc in the first stream, i.e. sequentially, starting with the first sub-packet, and the sub-packets are sent n, n-1, ... etc in the second stream, i.e. sequentially, starting with the last sub-packet. In a further embodiment, these two streams are sent almost simultaneous. The two streams can be sent over the same carrier, for instance using conventional multiplexing techniques, or they can be sent over entirely different carriers, for instance cable modem and telephone line. The digital data can also be of another form instead of the now-used binairy data.

Brief description of the drawings

The invention is further elucidated on the basis of the figures that are exemplary embodiments of the invention. However, the invention is not limited to said exemplary embodiments. Shown in the figures is:

Figure 1 the splitting of one signal into two signals and the adding together at the receiver;

Figure 2 the receipt of a split signal by a receiver from two physically

separated sources,

Figure 3 the provision of two receivers with one signal,

Figure 4 the provision of three receivers with one signal,

Figure 5 an alternative for the situation of figure 4,

Figure 6 an example of the relation between bandwidth and the quantity of signal,

Figure 7 a second example of the relation between bandwidth and quantity of signal,

Figure 8A-8C a receiving device,

Figure 9A-9C a sending device,

Figure 10A-10D a device which receives and almost simultaneously sends, and

Figure 11 the process of sending and receiving.

Description of preferred embodiments

Figure 1 shows a situation in which a signal 5 *in a conventional manner* enters a receiver 3. The receiver 3 splits the signal, or each data packet from which the signal has been built up, into two streams 1 and 2 to forward it to receiver 4. Stream 1 is the signal sent from the front, that means the first bit of the data packet or the signal is sent first, then the second, etc. Stream 2 is the signal 5 or a data packet thereof, but then backwards, that means first the last bit is sent then the last but one, etc.

Therefore, as both streams make up the entire stream, the two streams may be considered as complementary streams.

Receiver 4 simultaneously fills its data buffer from the front with signal 1 and from the rear with signal 2. This can take place by means of a computer program, but can also be implemented hardware-wise. When the buffer is full, that means the complete signal or data packet has been received, receiver 4 sends a signal to receiver/transmitter 3 that the buffer is full, that means that the signal has been received. It is of course also possible that the receiver 4 keeps sending a signal to receiver/transmitter 3 until the buffer is full, or just closes down the connection when the buffer is full, or sets the port at high or low.

The principle depicted in figure 1, can also be used in figure 2 with 2 sources 3, 3' that transmit to a receiver 4. In that case the sources 3, 3' receive the entire (or already split) signal or data packet 5, and each send a partial signal 1 or 2, respectively, to receiver 4. This offers advantages when the transmission capacity of sources 3 and 3' is lower than the receiving capacity of receiver 4. The receiver 4 may even forward the reconstructed signal or data packet 6 in its entirety again.

Figure 3 shows an example of the method according to the present invention, wherein one source 3 splits a signal or data packet 5 into two complementary streams 1 and 2. The one stream is sent to receiver 4, the other stream to receiver 4'. Both receivers 4 and 4' send what is received to each other, so that both obtain a complete signal or data packet again. This offers advantages when the transmission capacity/bandwidth of 3 is limited, but/and the capacity between 4 and 4' is sufficient for exchange of data.

Figure 4 shows an example wherein a source 3 splits a signal or data packet 5 into two streams, wherein stream 1 is sent to receivers 4, 4",

and stream 2 to receiver 4'. Receiver 4' forwards its part of the signal or data packet to 4 and 4", whereas 4 forwards its part to 4'. In the optimal case all receivers 4, 4' and 4" will receive the entire signal or data packet in less time than usually needed in conventional point-to-point connections, or while using less bandwidth.

Figure 5 shows an example of the use of the method according to the invention, wherein a source 3 having limited data transfer capacity splits a signal or data packet 5 into two streams 1 and 2. Stream 1 is sent to receiver 4, stream 2 to receiver 4", and receivers 4 and 4" forward their part to receiver 4'. As a result three receivers have received the entire signal or data packet within less time usually needed to send the packet in its entirety to all three receivers, and the bandwidth used is smaller.

Figures 6 and 7 show calculation examples wherein in case of figure 6 the available bandwidth from transmitter 3 to receivers 4 and 4' is almost the same. In that case receivers 4 and 4' will have received the data in 50% of the usually necessary time, and this is also the load for transmitter 3, seen in bandwidth, only 2 times 50% in total instead of 2 times 100%. Thus, digital data is sent without any overhead (transmitter 3 may stop sending when all data is sent) and very fast.

In the calculation example of figure 7 the connection is a-synchronous. Receiver 4' receives 91% of the total data packet, and receiver 4 receives 11% of the total data packet. The nett result is that transmitter 3 in the end only needs to send the total data packet once. Additionally there will be a small gain of speed in this unfavourable case. In this case, the sending capacity from transmitter 3 to receiver 4, as well as the sending capacity from receiver 4 to receiver 4'. Due to very lowtransmission capacity from receiver 4 to receiver 4' (or delays) only 9 % instead of 11 % could be transmitted from receiver 4 to receiver 4'.

In figure 8A-8C, the process of receiving a digital data packet split into two streams accoding to the present invention is shown. In figure 8A, two streams 21 and 22 are received and put into data buffer 20. The first part of the received data from stream 21 is put in place 1, the first part of received data from stream 22 is put in place n. I figure 8B, an intermediate step is shown. In this case, via stream 21 the 4th data part is received and is put into buffer at location 4, while via stream 22 also another (n-i th) data part is recieved. Despite the fact that the two streams are not equally fast, no overhead is needed.. In figure 8C, the two streams meet, the buffer is full. This will trigger the device to stop receiving, or send a signal that the digital data packet is complete.

In figures 9A-9C, the sending side is shown. In figure 9A, data buffer 23 is full of data. The device starts taking data from data buffer, starting at the front of the data buffer 23, and starts sending the data. At the same time, the device starts taking data from the back of databuffer 23, position n, and sends this data. In figure 9B, taken some time later, it can be seen that the device takes data part 3 from data buffer 23 and sends it. At the same time, from the rear end of the data buffer the fifth data part is teken and send. The device sequentially takes the next, etc, so the next data parts would be number 4 and the 6th from the back. In figure 9C, the last data parts are taken. Again, it can be seen that sending is swift, and without complex overhead.

Figures 10A-10D show the situation where there is a device which both sends and receives according to the present invention. This time, data buffer 26 is empty, The devive starts receiving data parts via streams 28 and 29, stream 28 is put at the first place, and stream 28 will subsequently fill data buffer 26 from the front to the back. The first data part received from stream 29 will be put into the last position, n, of the data buffer 26. Subsequently, stream 29 will fill data buffer 26 from the back to the front, until the buffer is full.

At the same time, the device starts sending data parts via streams 30 and 31. Stream 30 starts from the front of the data buffer and stream 31 starts from the back of the data buffer.

In figure 10C, it can be seen that in this case, the device receives data parts faster than it sends them. In figure 10C, all the data parts are received: the two streams 28 and 29 meet. The device keeps sending via streams 30 and 31. In figure 10D, the streams 30 and 31 meet, and the device can stop sending. This is all possible with a minimum on overhead.

In figure 11, The process of sending and receiving are depicted in one figure. Apparatus 40 having data buffer 24 sends two streams of data 44 and 45. Apparatus 41 receives these two streams as streams 46 and 47, and puts the data in data buffer 43. Again, the sending apparatus 40 has two streams 44 and 45, one starting at the front and one at the back of data buffer 42. Receiving apparatus 41 receives two streams, and places one stream in the front, and one in the back of data buffer 43.

For that matter the method according to the invention can for instance be used in GSM or other cordless telephony. A conversation or a data stream can then be divided into packets which, in accordance with the method according to the invention, can be sent. Use can then also be made of the available bandwidth: each data stream can be sent over another band, so that optimal use is made of the available bandwidth.

It is to be understood that the above description is included to illustrate the operation of the preferred embodiments and is not meant to limit the scope of the invention. The scope of the invention is to be limited only by the following claims. From the above discussion, many variations will be apparent to one skilled in the art that would yet be encompassed by the spirit and scope of the present invention.

Claims

1. A method for multiplexing digital data, wherein a packet of digital data, said packet of digital data having a length, a front end and a rear, is simultaneously sent from front to rear and from rear to front.

2. The method according to claim 1, wherein a first device sends the data from front to rear to a third device, and a second device sends the same data from rear to front to the third device.

3. The method according to claim 2, wherein the third device places the data in a data buffer the size of the packet, and sends a signal to the first and second device when either the buffer is full, or stops sending confirmations until the buffer is full.

4. The method according to claim 1, wherein a first device sends data from front to rear to a second device, and simultaneously backwards to a third device.

5. The method according to claim 4, wherein the second device and third device immediately at receipt forward the data they received from the first device to each other.

6. The method according to claim 5, wherein the second and third device have been provided with a data buffer the size of the packet, wherein the received data are placed in the data buffer and the first and second device send a signal to the first device when the respective data buffer is full.

7. A method for multiplexing digital data, wherein a packet of digital data having a length, a beginning and an end, is simultaneously sent from the beginning of the packet towards the end and from the end towards the beginning.

8. The method according to claim 7, wherein a first device sends the data from beginning towards the end to a third device, and a second device sends the same data from the end towards the beginning to the third device.

9. The method according to claim 8, wherein the third device places the data in a data buffer the size of the packet, and sends a signal to the first and second device when either the buffer is full, or stops sending confirmations until the buffer is full.

10. The method according to claim 7, wherein a first device sends data from the beginning to the end to a second device, and simultaneously backwards to a third device.

11. The method according to claim 10, wherein the second device and third device immediately at receipt forward the data they received from the first device to each other.

12. The method according to claim 11, wherein the second and third device have been provided with a data buffer the size of the packet, wherein the received data are placed in the data buffer and the first and second device send a signal to the first device when the respective data buffer is full.

13. A method for multiplexing digital data, wherein simultaneously a packet of digital data is sent by electronical or optical means and the same packet of digital data is sent backwards by electronical or optical means.

14. The method according to claim 13, wherein a first device sends the packet of digital data from front to rear to a third device, and a second device sends the same data backwards to the third device.

15. The method according to claim 14, wherein the third device places the data in a data buffer the size of the packet of digital date, and sends a signal to the first and second device when either the buffer is full, or stops sending confirmations until the buffer is full.

16. The method according to claim 13, wherein a first device sends data from front to rear to a second device, and simultaneously backwards to a third device.

17. The method according to claim 16, wherein the second device and third device immediately at receipt forward the data they received from the first device to each other.

18. The method according to claim 17, wherein the second and third device have been provided with a data buffer the size of the packet, wherein the received data are placed in the data buffer and the first and second device send a signal to the first device when the respective data buffer is full.

19. A method for sending a data packet to a first device in an ad-hoc data network of devices, wherein the devices have been provided with a data processing unit, a data buffer and software having receiving routines for receiving data packets from at least two transmitting devices in the data network, wherein at least two other devices in the network simultaneously send data sub-packets together making up said data packet, at least one device starting with the front sub-packet and sequentially taking the next sub-packet, and at least one devive starting with the last sub-packet and sequentially taking a previous one, to said first device, which first device

adds together these data sub-packets to form said data packet.

20. The method according to claim 19, wherein the software has further been provided with transmission routines for transmitting data packets, received from the transmitting device or devices in the data network to at least one receiving device that is connected to the data network, independent of the transmitting device or devices.

21. A method for receiving digital data, wherein a device provided with data storage means creates a data buffer in the data storage means the size of a packet of digital data, and simultaneously receives a first stream of digital data and receives a second stream of digital data, wherein the device fills the data buffer from front to rear with the first stream of digital data and fills the data buffer from rear to front with the second stream of digital data.

22. The method according to claim 21, wherein the device informs the source or sources of the streams of digital data when a data buffer is full.

23. A method for sending digital data, wherein a device provided with data storage means creates a data buffer in the data storage means, stores digital data in the data buffer, and from the front of the data buffer and the rear of the data buffer sends the digital data in two streams.

24. The method according to claim 23, wherein the device stops sending after receipt of a signal.

25. Software for sending a packet of digital data, comprising a first transmission routine for sending a first stream of digital data starting from the front of the packet of digital data and a second transmission routine for sending a second stream of digital data starting from the end of the packet of digital data

26. Software for receiving a packet of digital data, comprising a first receiving routine for receiving a first stream of digital data and a second recieving routine for simultaneously receiving a second stream of digital data, and a first storing routine for storing the first stream of digital data in a memory starting at the front of the memory and filling the memory towards the end, and a second storing routine for storing the second stream of digital data starting at the end of the memory and filling the memory towards the front, and a stop routine for ending the receiving of digital data when the memory is full.

27. An apparatus for sending a packet of digital data, comprising memory means for storing the packet of digital data, first sending means for sending a first stream of digital data, starting at the front of the memory means and second sending means for sending a second stream of digital data, starting at the end of the memory means.

28. An apparatus for receiving a packet of digital data, comprising memory means for storing the packet of digital data, first receiving means for receiving a first stream of digital data, and storing it in said memory means, starting from the front of the memory means, and second receiving means for receiving a second stream of digital data, and storing it in said memory means, starting from the back of the memory means.

29. Data carrier, provided with software according to any one of claims 25 or 26.

30. Device provided with software according to claim 25 or 26.

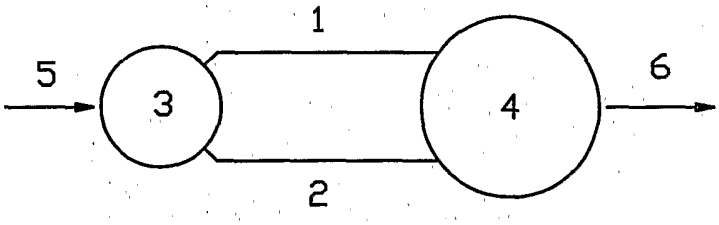

FIG. 1

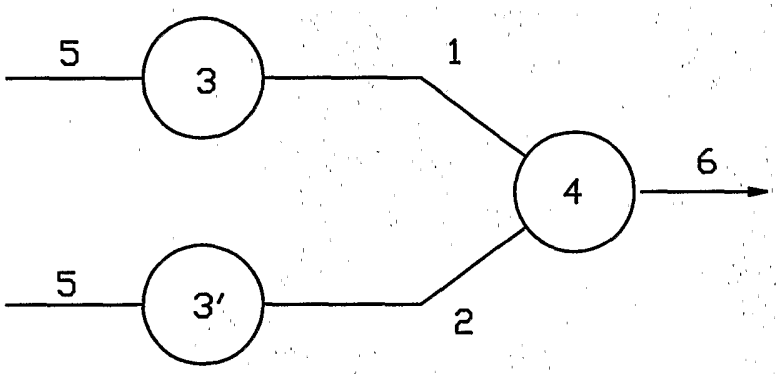

FIG. 2

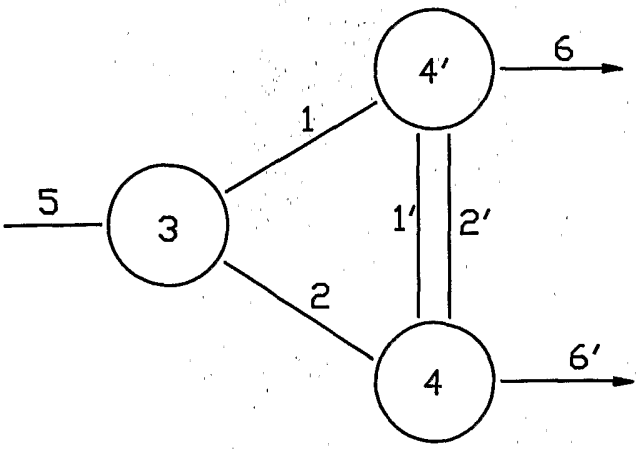

FIG. 3

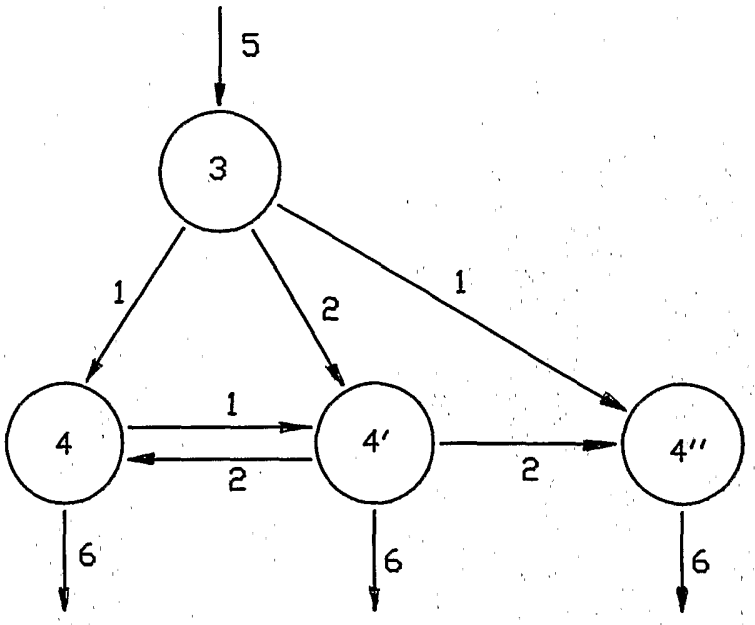

FIG. 4

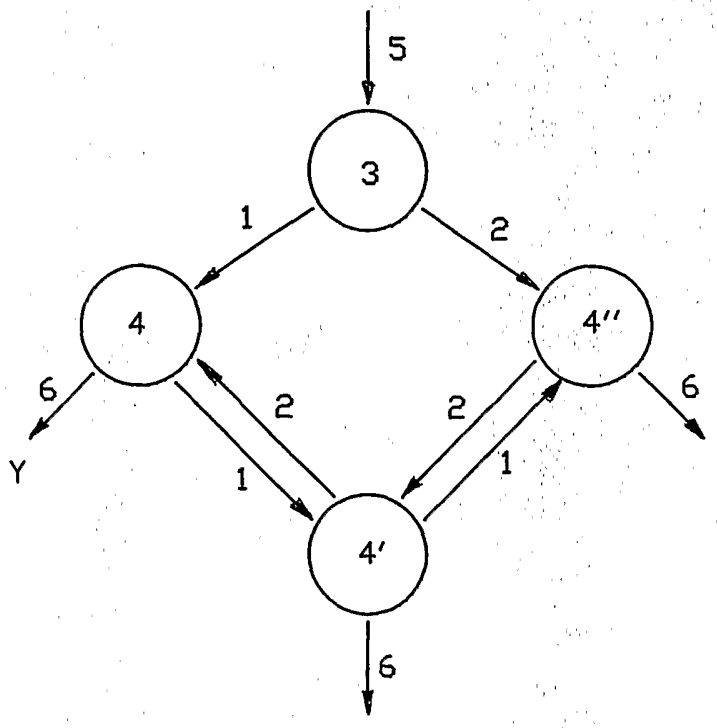

FIG. 5

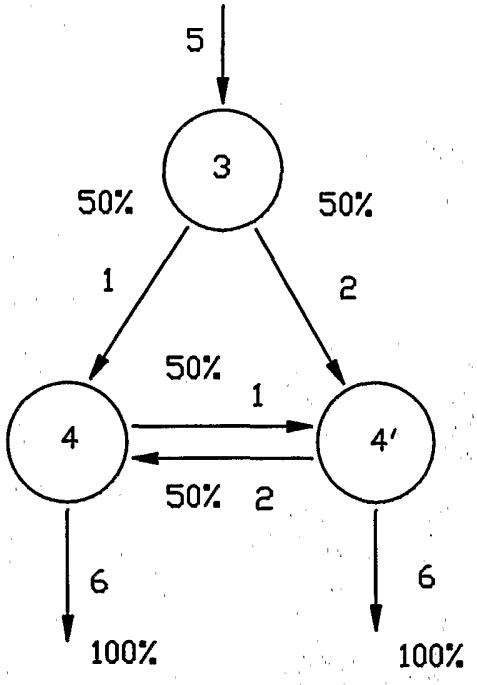

FIG. 6

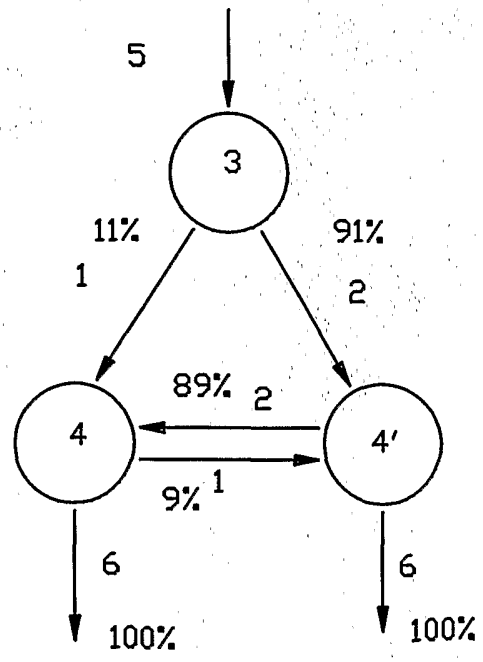

FIG. 7

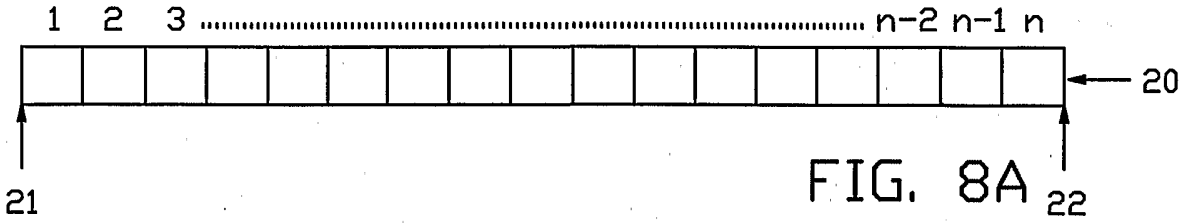

FIG. 8A

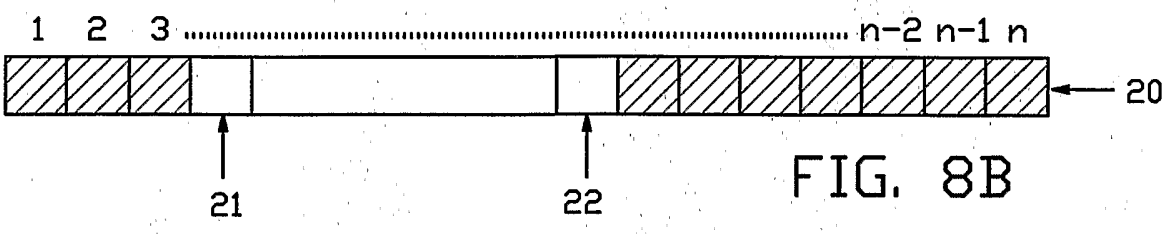

FIG. 8B

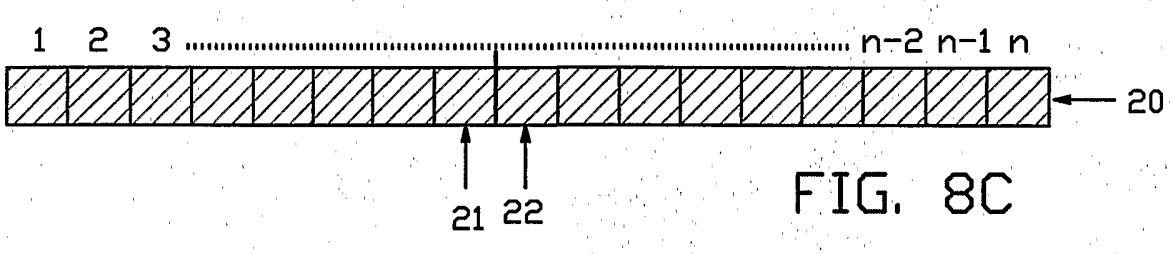

FIG. 8C

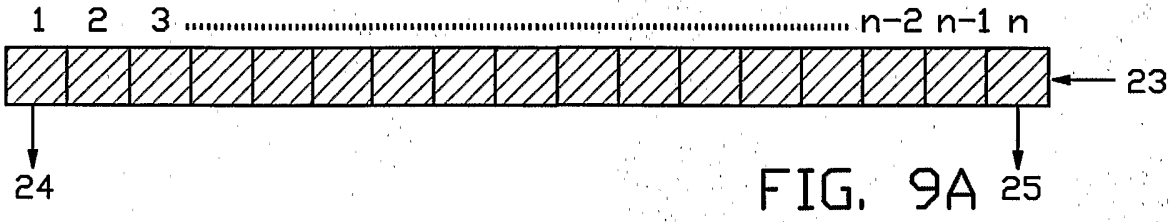

FIG. 9A

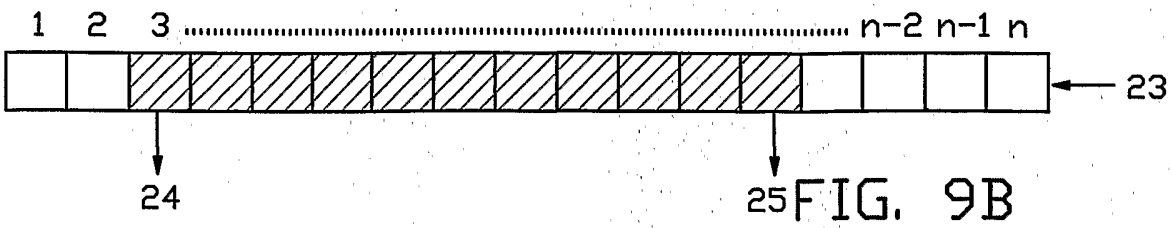

FIG. 9B

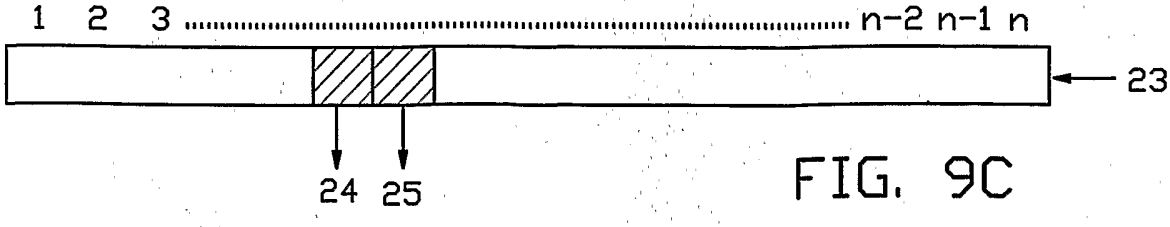

FIG. 9C

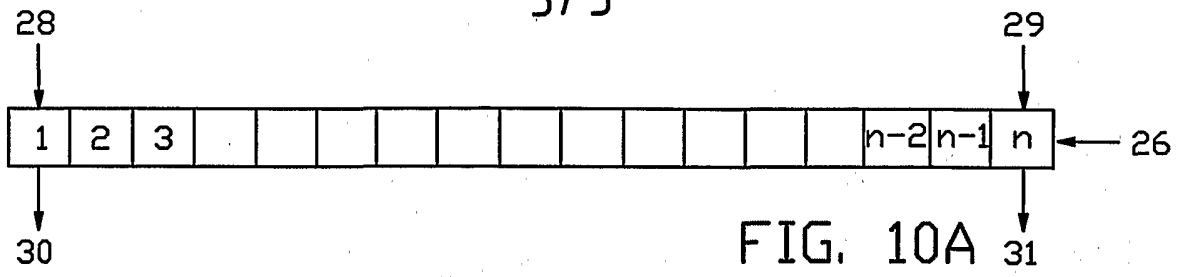

FIG. 10A

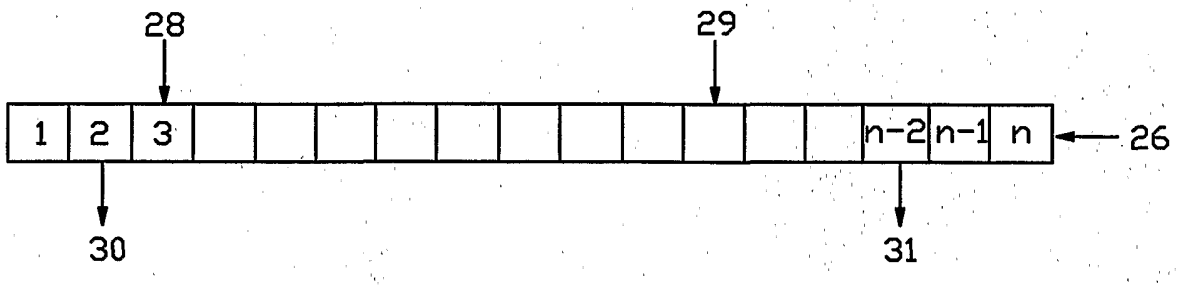

FIG. 10B

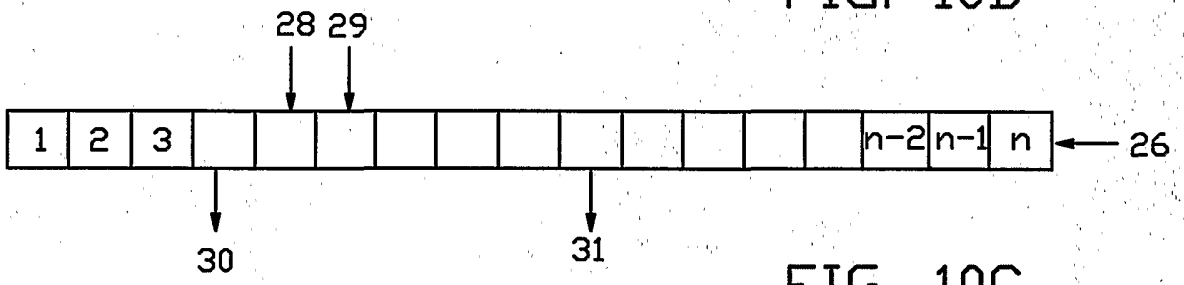

FIG. 10C

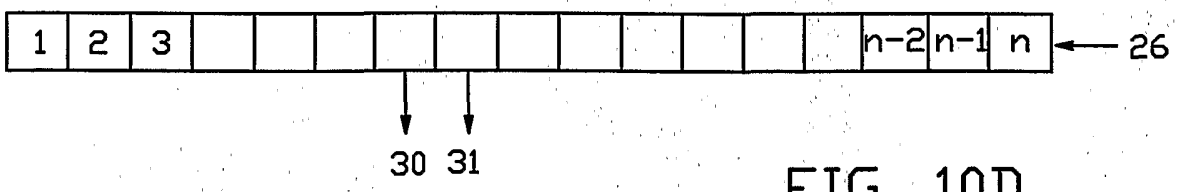

FIG. 10D

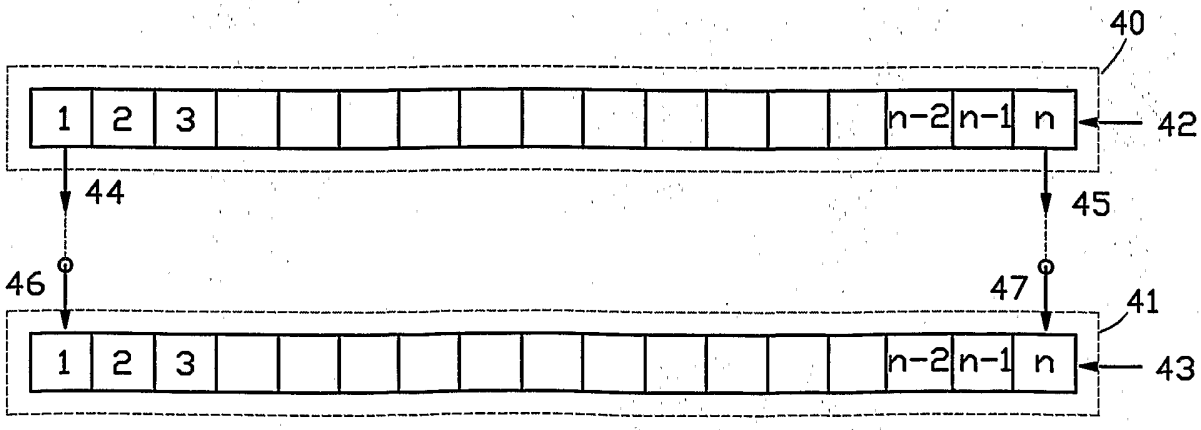

FIG. 11