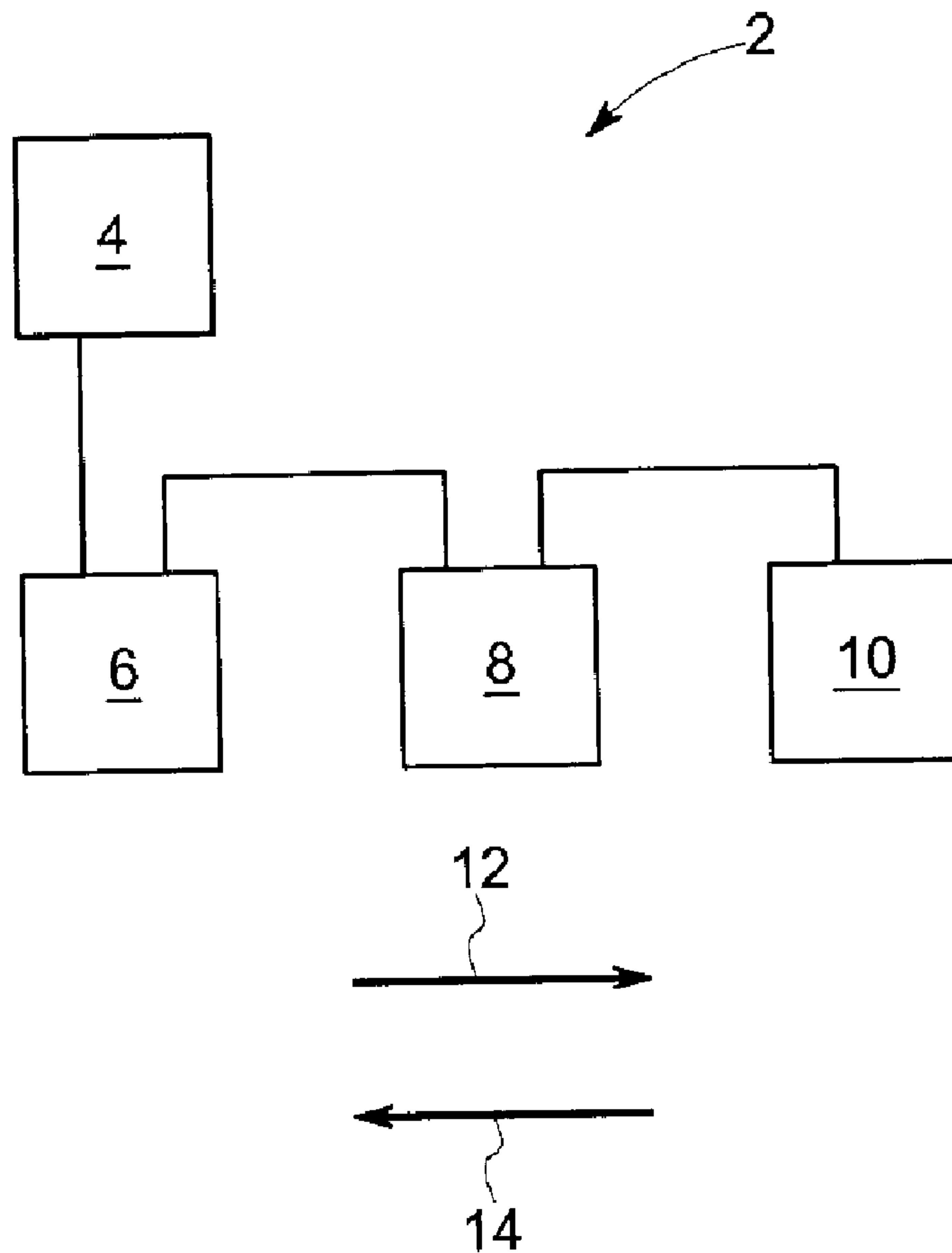




(22) **Date de dépôt/Filing Date:** 2014/04/29  
(41) **Mise à la disp. pub./Open to Public Insp.:** 2014/10/29  
(30) **Priorité/Priority:** 2013/04/29 (DE102013207826.9)

(51) **Cl.Int./Int.Cl. H04L 12/403** (2006.01),  
**H04L 12/951** (2013.01)  
(71) **Demandeur/Applicant:**  
GE ENERGY POWER CONVERSION GMBH, DE  
(72) **Inventeurs/Inventors:**  
OPITZ, THORSTEN, DE;  
WOTHE, FRANK, DE  
(74) **Agent:** CRAIG WILSON AND COMPANY

(54) **Titre : PROCEDURE DE FONCTIONNEMENT D'UN NOEUD ESCLAVE D'UN SYSTEME DE BUS NUMERIQUE**  
(54) **Title: METHOD FOR OPERATING A SLAVE NODE OF A DIGITAL BUS SYSTEM**



(57) **Abrégé/Abstract:**

A method for operating a slave node (8) of a digital bus system is described. The slave node (8) comprises two sending and receiving devices (16, 18). In the bus system, an input data frame is sent to a master node in input direction. The slave node (8)

**(57) Abrégé(suite)/Abstract(continued):**

receives the input data frame by means of the first sending and receiving device (18). The slave node (8) stores service data packets contained in the input data frame in a FIFO memory. The slave node (8) attaches at least one process data packet of its own, which is to be sent, to a last process data packet in the input data frame. The slave node (8) attaches the service data packets, which are stored in the FIFO memory, to the process data packet, which is now last, in the input data frame. The slave node (8) sends the input data frame, which was changed in this manner, to the next node in input direction by means of the second sending and receiving device (16).

270705

**METHOD FOR OPERATING A SLAVE NODE OF A DIGITAL BUS SYSTEM****ABSTRACT**

A method for operating a slave node (8) of a digital bus system is described. The slave node (8) comprises two sending and receiving devices (16, 18). In the bus system, an input data frame is sent to a master node in input direction. The slave node (8) receives the input data frame by means of the first sending and receiving device (18). The slave node (8) stores service data packets contained in the input data frame in a FIFO memory. The slave node (8) attaches at least one process data packet of its own, which is to be sent, to a last process data packet in the input data frame. The slave node (8) attaches the service data packets, which are stored in the FIFO memory, to the process data packet, which is now last, in the input data frame. The slave node (8) sends the input data frame, which was changed in this manner, to the next node in input direction by means of the second sending and receiving device (16).

270705

**METHOD FOR OPERATING A SLAVE NODE OF A DIGITAL BUS SYSTEM****DESCRIPTION**

The invention relates to a method for operating a slave node of a digital bus system.

A plurality of industrial digital bus systems and corresponding standards exists. Examples for this are PROFI BUS and Ethercat. PROFI BUS is a registered trademark of Phoenix Contact GmbH & Co. KG. Ethercat is a registered trademark of Hans Beckhoff.

WO 2009/021974 A2 discloses an Ethercat system for communication between master and slave systems. A communication cycle of the master comprises two data frames. The first data frame is suitable for retrieving data and the second data frame is suitable for sending data.

EP 1 223 710 A2 discloses a system for controlling actuators. The system comprises a bidirectional data bus between nodes, which are connected to the data bus. One node comprises two interfaces, wherein data, which are to be transmitted, are sent through both interfaces.

WO 03/054644 A2 discloses a method for transmitting data to a serial bus between at least one active bus subscriber and at least one passive bus subscriber.

For switching high-power semiconductor switching elements, which are connected to one another via an industrial bus system, a high synchronization of the individual bus subscribers and a high evaluation speed of the assigned

270705

sensor system must be ensured. A nearly simultaneous or highly synchronized switching, respectively, of the high-power semiconductor switching elements can only be ensured in this way and the high currents, which are to be switched, do not lead to the destruction of the system.

It is the task of the instant invention to provide a method for operating a slave node, which improves and speeds up the distribution of information.

The task is solved by means of a method for operating a slave node of a digital bus system according to claim 1.

In the case of the claimed method, the slave node stores service data packets, which are contained in an input data frame, in a FIFO memory. The slave node subsequently attaches at least one process data packet of its own, which is to be sent, to a last process data packet in the input frame. The slave node subsequently attaches the service data packets, which are stored in the FIFO memory, to the process data packet, which is now last. The input data frame, which was changed in this manner, is now sent to a master node in input direction.

Advantageously, the input data frame is utilized optimally through this and the master node receives the process data packets, which display the status of actuators, in particular high-power semiconductor switching elements, in presorted order. On the one hand, the input data frame, which passes through a plurality of slave nodes, is thus set up such that the process data packets are arranged in the front part of the input data frame. On the other hand, the position of the slave nodes in the bus system is considered. For example, the data packets of the slave node, which is furthest from the master node, are thus in first position, whereby the time, which the information requires to get through the bus system to the master node, is considered. After evaluating this information, a switching command can thus be sent immediately to the slave node or

270705

emergency measures, such as shut-off, can be initiated immediately when a malfunction has been determined.

Further features, potential applications and advantages of the invention follow from the below description of exemplary embodiments of the invention, which are explained by means of the drawing.

Figure 1 shows a schematically illustrated industrial bus system;

Figure 2 shows a schematic setup of a slave node;

Figure 3 shows the schematic setup of an input data frame or output data frame;

Figure 4 shows a detailed setup of a slave node in a schematic manner;

Figure 5 shows a schematically illustrated flow chart; and

Figure 6 shows a schematically illustrated communication process in an input direction towards a master node.

Figure 1 shows a schematically illustrated setup of an industrial bus system 2, which comprises a master node 4 as well as a plurality of slave nodes 6, 8 and 10. The slave nodes 6, 8 and 10 are connected to one another in series via corresponding data lines. The master node 4 is arranged at one end of the slave nodes 6 and 10, which are connected to one another in series, wherein the master node is also connected to the slave node 6 via a corresponding data line.

Originating at the master node 4, an output data frame is sent in output direction 12 to the last slave node 10, which is located opposite the master node 4. Originating at the master node 4, the output data frame is initially sent to the slave node 6 for this purpose. The slave node 6 sends the received output data

270705

frame to the slave node 8. The slave node 8 sends the received output data frame to the slave node 10.

Originating at the last slave node 10, which is arranged at the opposite end of the master node 4, an input data frame is sent to the master node 4 in input direction 14 as answer to the output data frame.

As is not shown in Figure 1, the master node 4 can have a redundant design. For this purpose, a further master node is connected to the last slave node 10, wherein the further master node takes over the function of the last slave node and thus monitors the network traffic. If a breakdown of the master node 4 is identified, the further master node takes over the function thereof.

The master node defines a time period between the receipt of the output data frame and the sending of the input data frame for the last slave node 10.

By sending the output data frame, the master node 4 thus defines the communication via the industrial bus system 2. The sending of output data frames through the master node 4 can take place within a fixed or variable cycle time. The first slave node 6, which is arranged adjacent to the master node 4, provides its time signal to the other nodes 4, 8 and 10. To initialize the bus system 2, the run times between the adjacent slave nodes 6 and 8 as well as 8 and 10 are determined, whereby a synchronization of the clocks is attained in the slave nodes 6 to 10.

The slave nodes 6, 8, 10 are preferably designed in hardware or FPGA technology (FPGA stands for Field Programmable Gate Array). In contrast, the master node 4 is preferably designed as industry-pe comprising a real-time operating system.

Figure 2 shows the schematic setup of the slave node 8. The slave nodes 6 and 10 are set up identically. The slave node 8 comprises a first and a second

270705

sending and receiving device 16 and 18. The two sending and receiving devices are in each case designed as Gigabit-Ethernet interface, in particular according to the standards IEEE 802.3z or IEEE 802.3ab. A communication block 20 takes over the coordination between the two sending and receiving devices 16 and 18. Actuators, such as high-power semiconductor switching elements, as well as sensors, which are assigned to the actuators, are assigned in the application block 22 in a manner, which is not shown.

In the case of the last slave node 10, without the redundant master node, only one of the sending and receiving devices 16 and 18 is connected to a further node, for example to the slave node 8. The last slave node 10 recognizes that one of the sending and receiving devices 16 or 18 is open and is not connected to a further node, and thus takes over the afore-mentioned functions of the last slave node 10.

In particular, a high-power semiconductor switching element is assigned to the slave node 8 for switching sensors, wherein the high-power semiconductor switching element is switched as a function of output data, which are transmitted via the output data frame, and wherein the input data, which are created by the sensors, are sent to the master node 4 by means of the input data frame.

Figure 3 shows the setup of a data frame 24 in a schematic manner, wherein the data frame 24 represents input data frame as well as output data frame. The afore-mentioned Gigabit-Ethernet technology is used to send the data frame 24. This is why the data frame 24 is arranged in the payload of a Gigabit-Ethernet data frame 26. The Gigabit-Ethernet data frame 26 includes a head part 28, which is arranged upstream of the data frame 24, as well as a test part 30, which is arranged downstream from the data frame 24.

A broadcast address is always specified as MAC address in the head part 28 of the Gigabit-Ethernet data frame 26, because the network traffic can thus be

270705

observed in a simple manner, when a so-called packet sniffer is arranged between two network nodes. Prior to sending, the test part 30 is always calculated anew by a node. The Gigabit-Ethernet data frame 26 corresponds to the standard IEEE 802.3.

The data frame 24 consists of a head part 32 as well as of process data packets P, which connect to the head part 32, and of service data packets S, which connect to the process data packets. The service data packets S are optional. The head part 32, which in particular includes the clock time of the first slave node 6, is thus followed by a first section 34 of process data packets and by an optional second section 36 of service data packets S. The process data packets P serve to operate the actuators and to evaluate the sensors. The service data packets S serve to configure and update the bus system 2 and the nodes thereof.

Figure 4 shows a detailed setup of the slave node 8 in a schematic manner. Contrary to Figure 2, the communication block 20, originating at the application block 22, is divided into two parts 20a and 20b. Likewise, the first and the second sending and receiving device 16 and 18 are divided into the respective parts 16a, 16b and 18a and 18b.

In output direction 12, the receiving device 16b receives a Gigabit-Ethernet data frame 26 and leads at least the output data frame 24 to the communication block 20b. In output direction 12 away from the master node 4, the slave node 8 therefore receives an output data frame 24 by means of the sending and receiving device 16. The slave node 8 only has read accesses to the content of the output frame 24 and provides the content of the output data frame 24 to the application block 22. The output data frame 24 is sent through the slave node 8 to the next node in output direction 12 by means of the sending device 18b of the sending and receiving device 18.

270705

The receiving device 18a of the sending and receiving device 18 receives a Gigabit-Ethernet data frame 26 and provides the content of the input data frame 24 to the communication block 20a, wherein the communication block 20a is allowed to access the frame 24. Data provided by the application block 22 are in particular inserted into the input data frame 24. The communication block 20a provides the input data frame 24, which was changed in this manner, to the sending device 16a of the sending and receiving device 16 for sending purposes, wherein the sending device 16a creates a new Gigabit-Ethernet data frame 26, the payload of which includes the changed input data frame 24. The input data frame 24 is sent through the slave node 8 to the next node in input direction 14 by means of the sending device 16a of the sending and receiving device 16.

As suggested by the dashed lines around the application block 22, the processing of the output data frame and of the input data frame in the transport-oriented layers of the slave node 8 is preferably carried out parallel to one another and independent from one another. The elements above and below the application block 22 are part of the transport-oriented layers of the slave node 8.

Figure 5 shows a schematically illustrated block diagram 38. The receiving device 18a provides the input data frame 24 to the communication block 20a. Service data packets S contained in the input data frame 24 are read in a block 40 in a first step and are stored in a FIFO memory 42. According to arrow 44, at least one further process data packet P is provided to the block 40. In a further step, the block 40 adds the process data packet P, which is provided according to arrow 44, to a last process data packet P in the input data frame 24. The slave node 8 thus attaches at least one process data packet P of its own, which is to be sent to a last process data packet P in the input data frame. However, if a process data packet P is not yet available in the received input data frame 24, the further process data packet P is the first process data packet in the input data frame 24, which is to be sent.

270705

According to arrow 4S, the slave node 8 provides a service data packet S, which is to be sent, to the block 46. The block 46 adds the service data packet S, which is provided by the slave node 8, to the service data packets S from the FIFO memory 42, which are already in the input data frame 24, which is to be sent. In the alternative, the service data packet S, which is provided according to arrow 4S, can also be supplied to the FIFO memory 42 after filling the FIFO memory 42 with the service data packets S from the received input data frame 24.

In a block 46, the service data packets S, which are stored in the FIFO memory 42, are attached to the process data packet P, which is now last. The input data frame 24 has now been changed such that all of the process data packets are located in a first part 34, as is shown in Figure 3, and all of the service data packets are located in a second part 36, which follows the first part 34. The input data frame 24, which was changed in this manner, is sent to the next node in input direction 14 by means of the sending device 16a.

A maximum size can be determined for the content of the input data frame 24. According to block 40, the slave node 8 attaches its own process data packet, which is to be sent, to the last process data packet in the input data frame 24. According to the block 46, the slave node 8 attaches a first number of the service data packets S, which are stored in the FIFO memory 42, to the process data packet P, which is now last, such that the maximum size for the content of the input data frame is not exceeded. The slave node 8 sends the input data frame 24 to the next slave node 6 in input direction 14 by means of the sending device 16a. A second number of service data packets S, which remained in the FIFO memory 42, is sent in a next input data frame 24.

In the alternative or in addition, the master node 4 can ensure that the maximum size for the content of the input data frame 24 is not exceeded in that the master node 4 provides for a threshold value for the content of the input data frame 24 and so that the master node 4 only requests so many process data packets from

270705

the slave nodes 6, 8 by means of the output data frame 24 so that the threshold value for the content of the input data frame 24, which follows as answer to the sending of the output data frame 24 through the master node 4, is not exceeded.

Figure 6 shows the function of the communication block 20a in an exemplary and schematic manner. The slave node 10 thus sends the output data frame 24 within a Gigabit-Ethernet data frame 26 to the slave node 8. When receiving through the slave node 8, the output data frame 24 receives the head part 32 as well as a process data packet P10 and a service data packet S10.

The slave node 8 receives the input data frame 24 from the slave node 10. The slave node 8 must send a process data packet P8 and a service data packet S8. According to the communication block 20a from Figure 5, the process data packet P8 is added downstream from the process data packet P10. The service data packet S10 is buffered in the FIFO memory 42 and is inserted in the block 46 downstream from the process data packet P8. The service data packet S8 is inserted downstream from the service data packet S10. The same method is carried out in the slave node 6, wherein the slave node 6 used the process data packet P6 and the service data packet S6.

270705

**Claims**

1. A method for operating a slave node (6; 8) of a digital bus system (2), wherein the slave node (6; 8) is designed in particular for operating a high-power semiconductor switching element, wherein the slave node (6; 8) comprises two sending and receiving devices (18, 16), wherein an input data frame (24) is sent to a master node (4) in the bus system (2) in an input direction (14), characterized in that the slave node (6; 8) receives the input data frame (24) by means of the first sending and receiving device (18), that the slave node (6; 8) stores service data packets (S10, S8; S10), which are contained in the input data frame (24), in a FIFO memory (42), that the slave node (6; 8) attaches at least one process data packet (P6; P8) of its own, which is to be sent, to a last process data packet (P8; P10) in the input data frame (24), that the slave node (6; 8) attaches the service data packets (S10, S8; S10), which are stored in the FIFO memory (42), to the process data packet (P6; P8), which is now last, in the input data frame (24), and that the slave node (6; 8) sends the input data frame (24), which was changed in this manner, to the next node (4; 6) in input direction (14) by means of the second sending and receiving device (16).

2. The method according to claim 1, wherein a maximum size is determined for the content of the input data frame (24), wherein the slave node (6; 8) attaches its own process data packet (P6; P8), which is to be sent, to the last process data packet (P8; P10) in the input data frame (24), wherein the slave node (6; 8) attaches a first number of the service data packets (S8, S10; S10), which are stored in the FIFO memory (42), to the process data packet (P6; P10), which is now last, in the input data frame (24), such that the maximum size for the content of the input data frame (24) is not exceeded, wherein the slave node (6; 8) sends the input data frame (24) to the next node (4; 6) in input direction (14), and wherein a second number of service data packets (S), which remained in the FIFO memory (42), is sent in a next input data frame (24).

270705

3. The method according to claim 1 or 2, wherein, in an output direction (12) away from the master node (4), the slave node (6; 8) receives an output data frame (24) by means of the second sending and receiving device (16), wherein the slave node (6; 8) only has read access to the content of the output data frame (24), wherein the slave node (6; 8) sends the output data frame (24) by means of the first sending and receiving device (18), and wherein the processing of the output data frame (24) and of the input data frame (24) takes place in the transport-oriented layers of the slave node (6; 8), parallel to one another and independent from one another.

4. The method according to any one of the preceding claims, wherein the input data frame (24) and/or the output data frame (24) contains a head part (32), subsequent process data packets (P), and service data packets (S), which are attached to the process data packets (P).

5. The method according to the preceding claim, wherein the process data packets (P) are arranged in the input data frame (24) such that, originating at the slave node (10), which is located opposite the master node (4), a process data packet (P10) is thus in first position downstream from the head part (32) and the process data packets (P8, P6) of the further slave nodes (8, 6) are subsequently arranged in the order of the slave nodes (6, 8) in the direction of the master node (4).

6. A slave node (6; 8) of a digital bus system (2), which is designed for carrying out a method according to any one of the preceding claims.

7. The slave node (6, 8) according to claim 6, wherein the first and second sending and receiving device (18, 16) are in each case designed as Gigabit-Ethernet interface, in particular according to IEEE 802.3z or IEEE 802.3ab.

270705

8. The slave node (6; 8) according to claim 6 or 7, wherein a high-power semiconductor switching element and a sensor are assigned to the slave node (6; 8), wherein the high-power semiconductor switching element is switched as a function of output data, which are transmitted by means of the output data frame (24), and wherein the input data, which are created by the sensor, are sent to the master node (4) by means of the input data frame (24).

9. A digital bus system (2) in particular for operating high-power semiconductor switching elements, comprising a master node (4), at least one slave node (6; 8) according to any one of claims 6 to 8, and a last slave node (10), which is arranged at the opposite end of the master node (4).

10. The digital bus system according to claim 9, wherein the slave nodes (6, 8, 10) are connected to one another in series, wherein the master node (4) is arranged at an end of the slave nodes (6, 8, 10), which are connected to one another in series, wherein, originating at the master node (4), the output data frame (24) is sent to the last slave node (10), which is located opposite the master node (4), in output direction (12), and wherein, originating at the last slave node (10), which is arranged at the opposite end of the master node (4), an input data frame (24) is sent to the master node (4) in input direction (14) as answer to the output data frame (24).

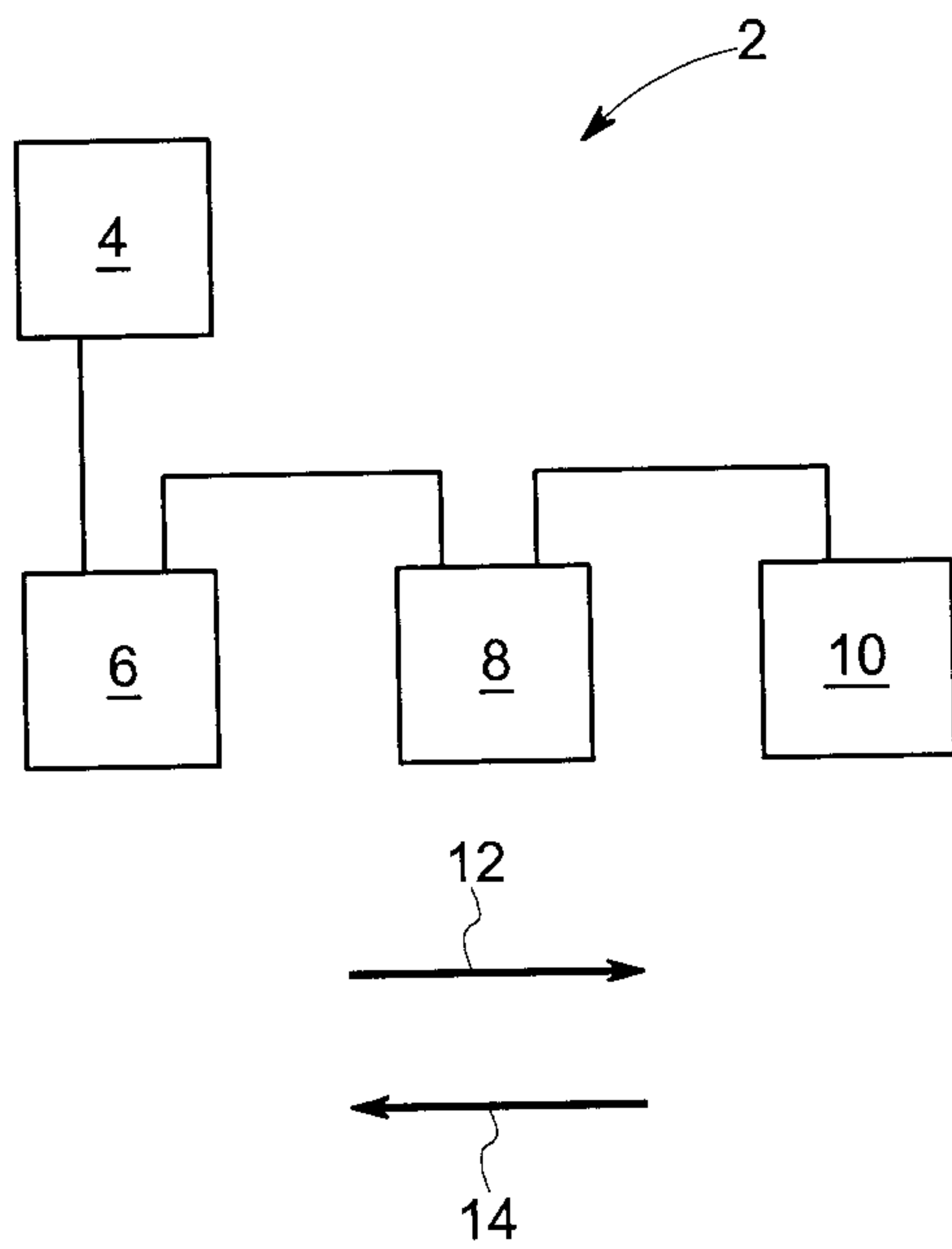


FIG. 1

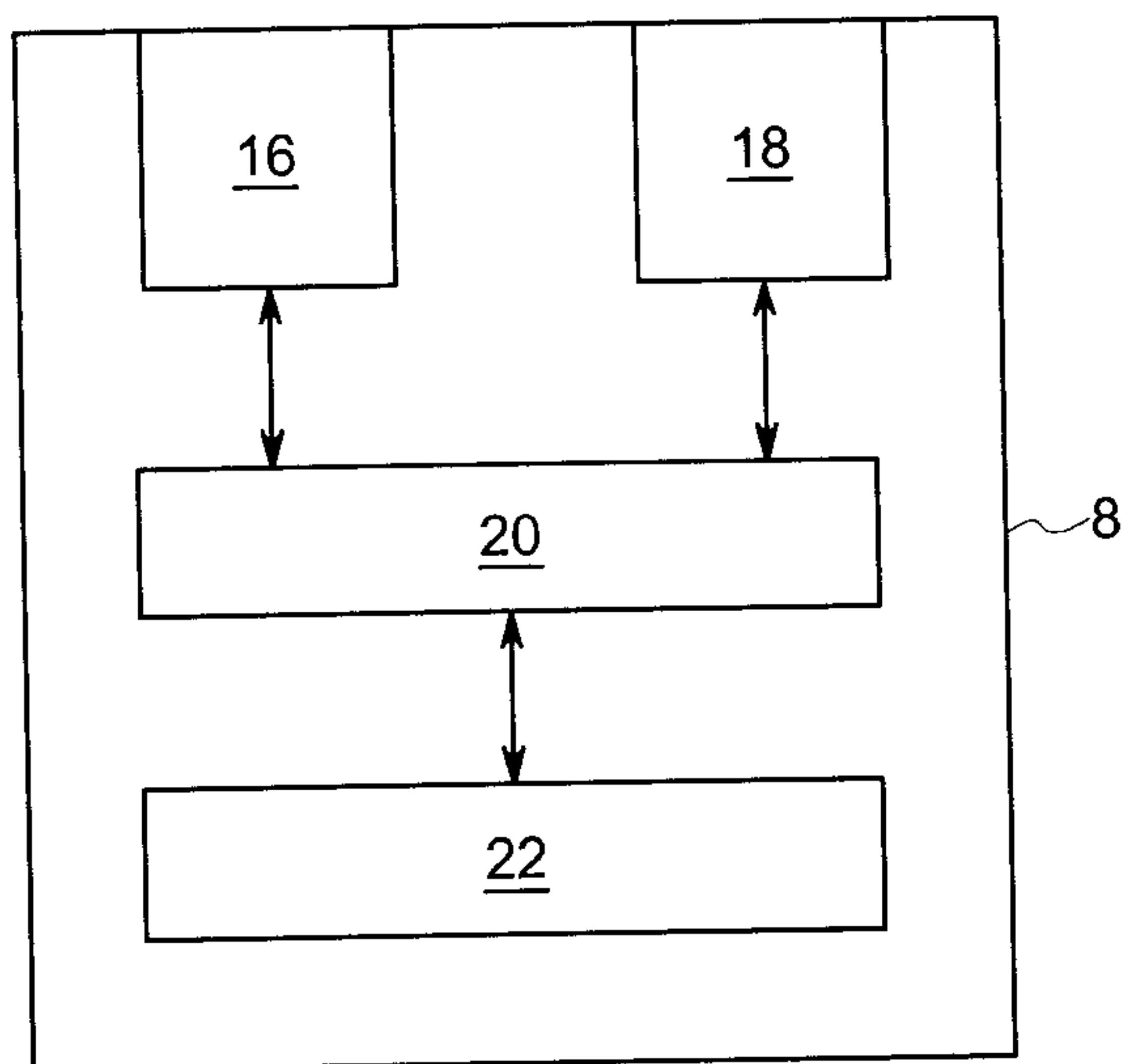


FIG. 2

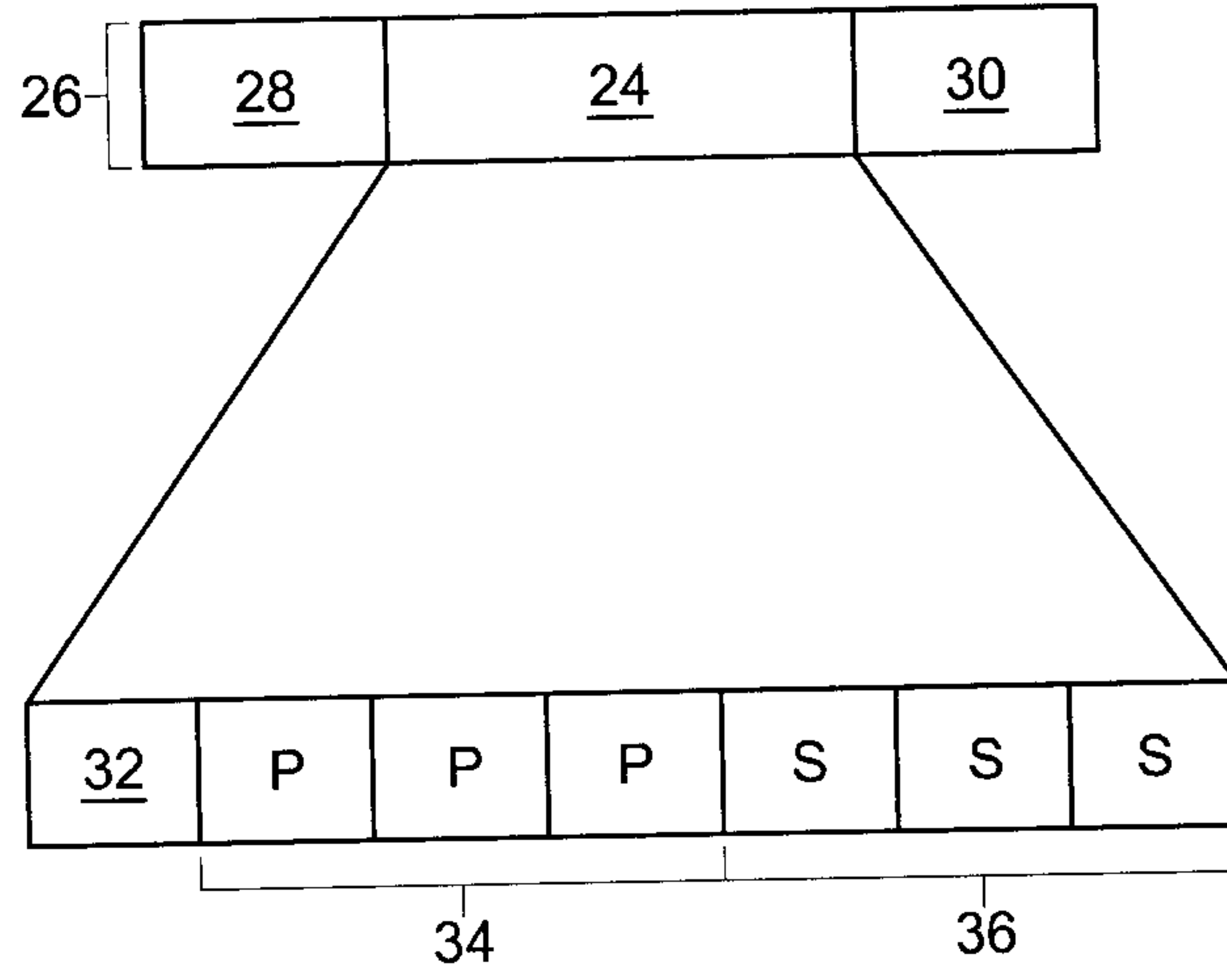


FIG. 3

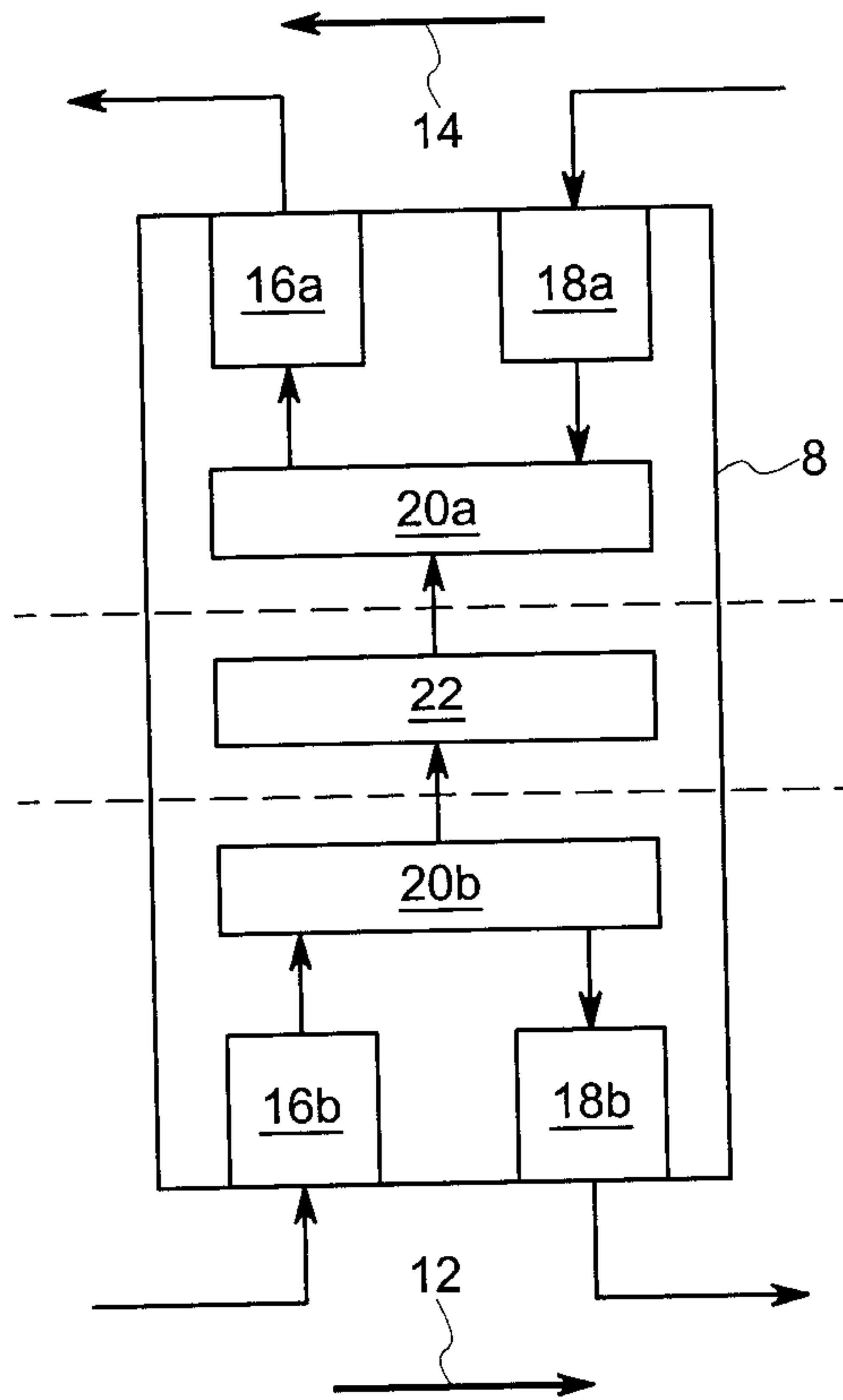


FIG. 4

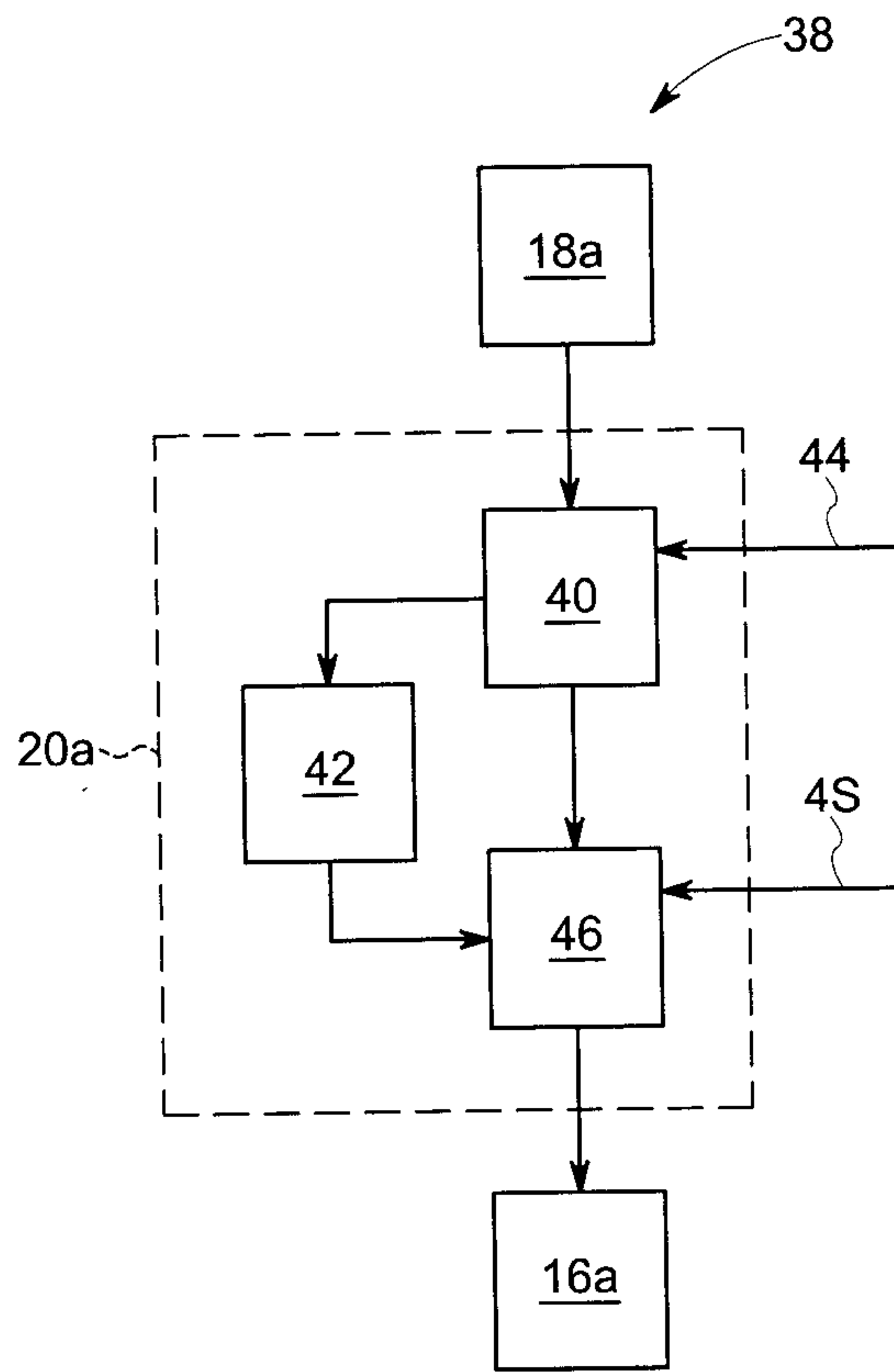


FIG. 5

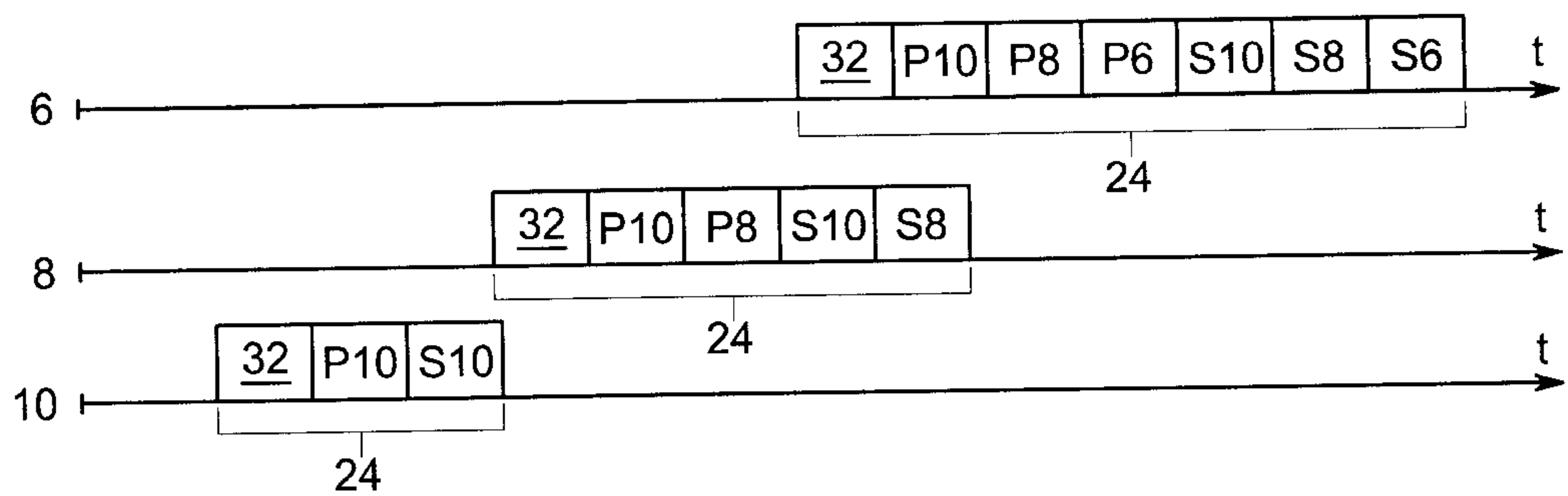


FIG. 6

