

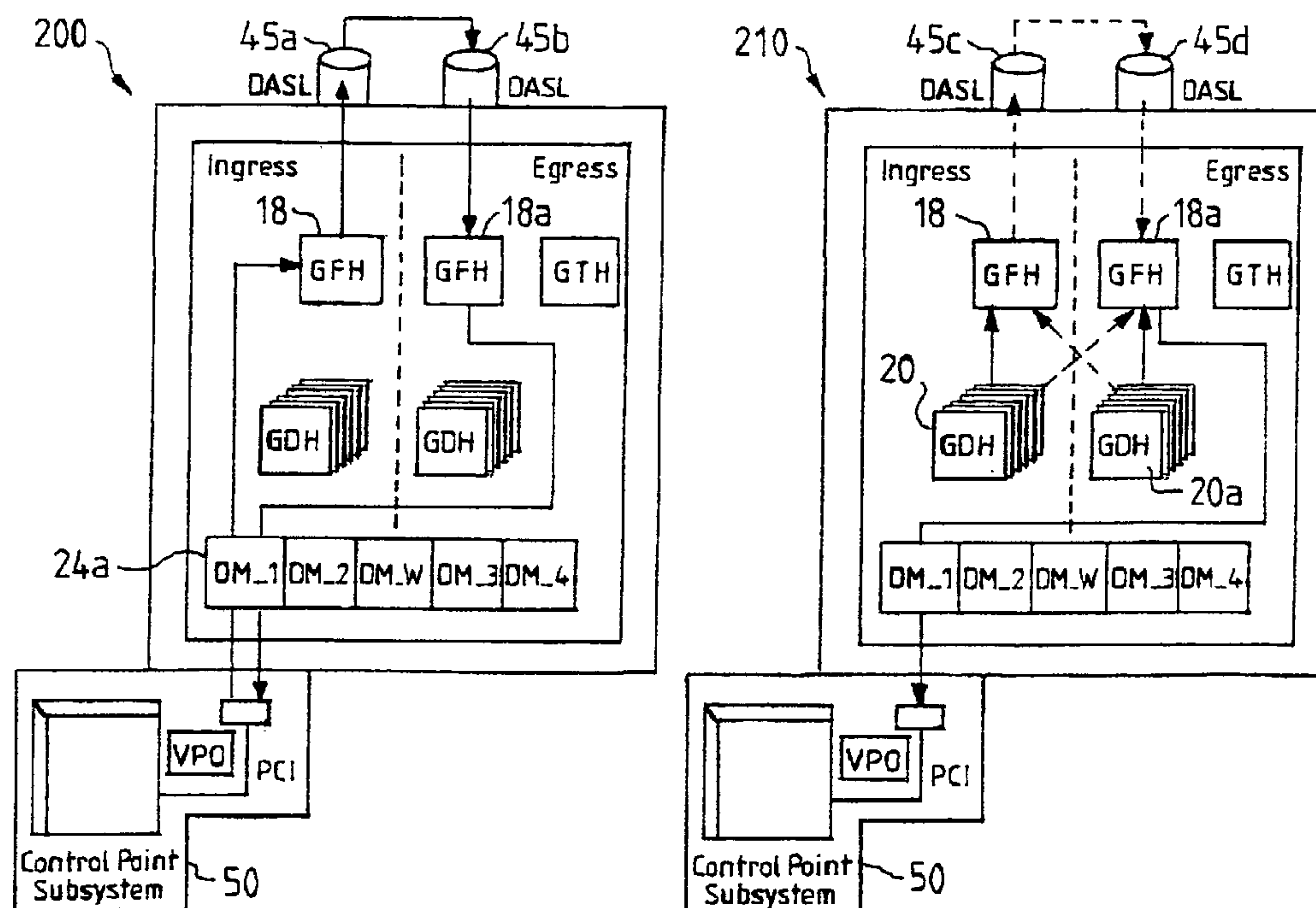


(86) Date de dépôt PCT/PCT Filing Date: 2001/04/10
(87) Date publication PCT/PCT Publication Date: 2001/10/18
(85) Entrée phase nationale/National Entry: 2002/08/22
(86) N° demande PCT/PCT Application No.: GB 2001/001600
(87) N° publication PCT/PCT Publication No.: 2001/078308
(30) Priorité/Priority: 2000/04/11 (09/547,362) US

(51) Cl.Int.⁷/Int.Cl.⁷ H04L 12/24
(71) Demandeur/Applicant:
INTERNATIONAL BUSINESS MACHINES
CORPORATION, US
(72) Inventeurs/Inventors:
BASSO, CLAUDE, US;
GALLO, ANTHONY MATTEO, US;
HEDDES, MARCO, US;
HARIHARAN, SEETA, US;
RAO, SRIDHAR, US;
ROVER, SONIA KIANG, US
(74) Agent: BARRETT, B.P.

(54) Titre : PROCEDE POUR DEFINIR ET COMMANDER LE COMPORTEMENT GLOBAL D'UN DISPOSITIF
PROCESSEUR DE RESEAU

(54) Title: METHOD FOR DEFINING AND CONTROLLING THE OVERALL BEHAVIOR OF A NETWORK PROCESSOR
DEVICE



(57) **Abrégé/Abstract:**

A system and method for controlling overall behavior of a network processor device implemented in a network processing environment servicing a communications network. The method includes steps of receiving a guided control frame including one or more control functions for configuring various functional devices within the network processor with device control parameter data; a step of forwarding one or more control functions from a received control frame to a functional device within the network processor to be configured; and, executing the control functions as specified in the control frame. A novel control frame data structure and communications infrastructure is implemented whereby any network processor device operating in a distributed network processing environment may be controlled in accordance with executed control functions and device control parameter data.



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

(19) World Intellectual Property Organization
International Bureau(43) International Publication Date
18 October 2001 (18.10.2001)

PCT

(10) International Publication Number
WO 01/78308 A3(51) International Patent Classification⁷: **H04L 12/24**

(21) International Application Number: PCT/GB01/01600

(22) International Filing Date: 10 April 2001 (10.04.2001)

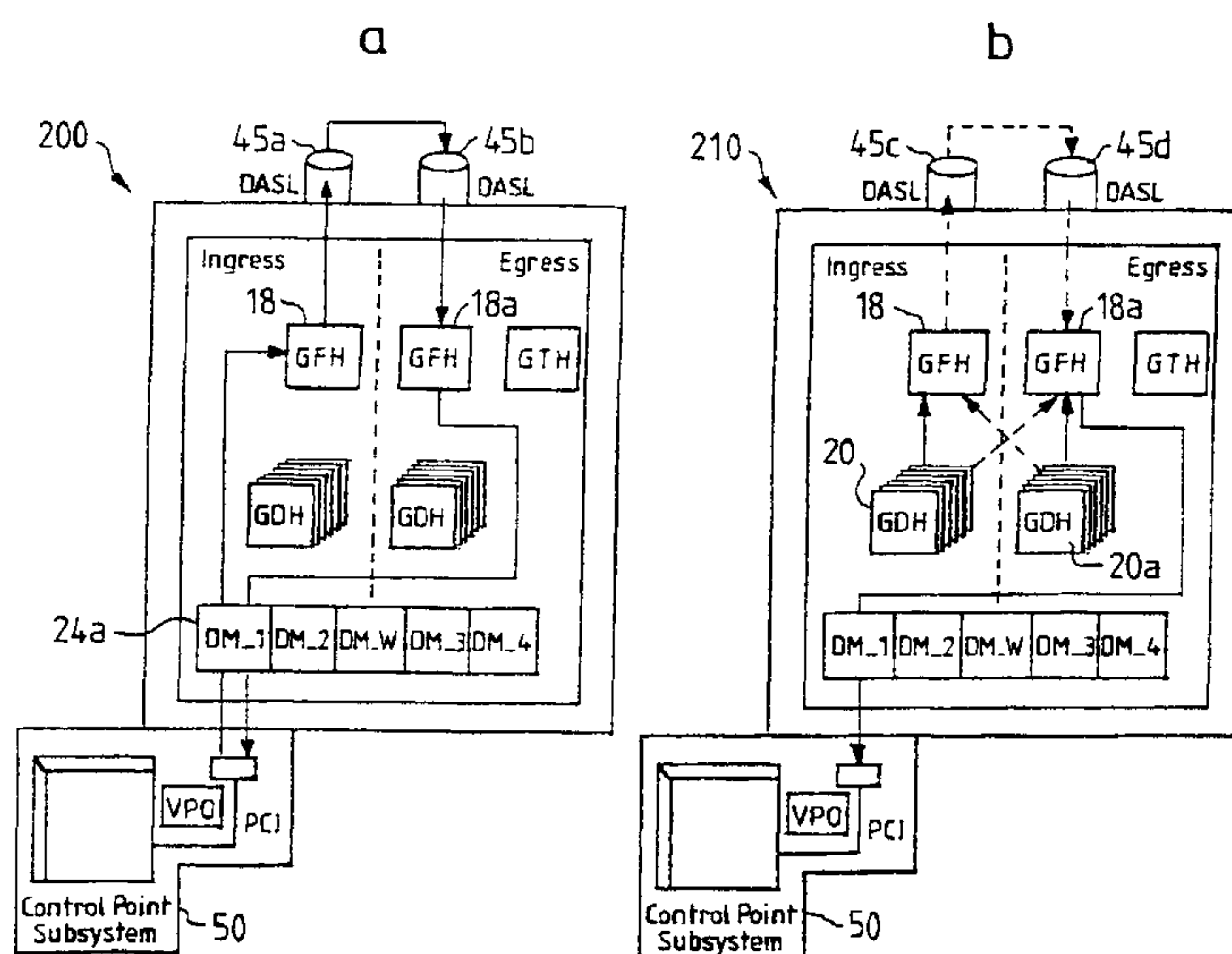
(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/547,362 11 April 2000 (11.04.2000) US(71) Applicant: **INTERNATIONAL BUSINESS MACHINES CORPORATION** [US/US]; New Orchard Road, Armonk, NY 10504 (US).(71) Applicant (for MC only): **IBM UNITED KINGDOM LIMITED** [GB/GB]; P.O. Box 41, North Harbour, Portsmouth, Hampshire PO6 3AU (GB).(72) Inventors: **BASSO, Claude**; 7604 Percy Court, Raleigh, NC 27613 (US). **GALLO, Anthony, Matteo**; 3308 Corsham Drive, Apex, NC 27502 (US). **HEDDES, Marco**; 4109 Grand Manor Court #308, Raleigh, NC 27615 (US).**HARIHARAN, Seeta**; 5020-104 Beaverbrook Road, Raleigh, NC 27612 (US). **RAO, Sridhar**; Apartment 204, 5020 Beaverbrook, Raleigh, NC 27612 (US). **ROVER, Sonia, Kiang**; 209 Parkside Circle, Chapel Hill, NC 27516 (US).(74) Agent: **BURT, Roger, James**; IBM United Kingdom Limited, Intellectual Property Law, Hursley Park, Winchester, Hampshire SO21 2JN (GB).(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: METHOD FOR DEFINING AND CONTROLLING THE OVERALL BEHAVIOR OF A NETWORK PROCESSOR DEVICE



(57) Abstract: A system and method for controlling overall behavior of a network processor device implemented in a network processing environment servicing a communications network. The method includes steps of receiving a guided control frame including one or more control functions for configuring various functional devices within the network processor with device control parameter data; a step of forwarding one or more control functions from a received control frame to a functional device within the network processor to be configured; and, executing the control functions as specified in the control frame. A novel control frame data structure and communications infrastructure is implemented whereby any network processor device operating in a distributed network processing environment may be controlled in accordance with executed control functions and device control parameter data.

WO 01/78308 A3

METHOD FOR DEFINING AND CONTROLLING
THE OVERALL BEHAVIOR OF A NETWORK PROCESSOR DEVICE

BACKGROUND OF THE INVENTION

5

Field of the Invention

This invention relates generally to network processor devices, and more specifically to an improved method for defining and controlling the overall behavior of a network processor.

10

Discussion of the Prior Art

In today's networked world, bandwidth is a critical resource. Increasing network traffic, driven by the Internet and other emerging applications, is straining the capacity of network infrastructures. To keep pace, organizations are looking for better technologies and methodologies to support and manage traffic growth and the convergence of voice with data.

15

20

Today's dramatic increase in network traffic can be attributed to the popularity of the Internet, a growing need for remote access to information, and emerging applications. The Internet alone, with its explosive growth in e-commerce, has placed a sometimes insupportable load on network backbones. It is also the single most important cause of increased data traffic volumes that exceed voice traffic for the first time. The growing demands of remote access applications, including e-mail, database access, and file transfer, are further straining networks.

25

30

The convergence of voice and data will play a large role in defining tomorrow's network environment. Currently, the transmission of data over Internet protocol (IP) networks is, in some countries, free. Because voice communications will naturally follow the path of lowest cost, voice will inevitably converge with data. Technologies such as Voice over IP (VoIP), Voice over ATM (VoATM), and Voice over Frame Relay (VoFR) are cost-effective alternatives in this changing market. However, to make migration to these technologies possible, the industry has to ensure quality of service (QoS) for voice and determine how to charge for voice transfer over data lines. The United States' Telecommunications Deregulation Act of 1996 further complicates this environment. This legislation will reinforce a symbiotic relationship between the voice protocol of choice, ATM, and the data protocol of choice, IP.

35

40

Integrating legacy systems is also a crucial concern for organizations as new products and capabilities become available. To preserve their investments in existing equipment and software, organizations demand solutions that allow them to migrate to new technologies without disrupting their current operations.

Eliminating network bottlenecks continues to be a top priority for service providers. Routers are often the source of these bottlenecks. However, network congestion in general is often mis-diagnosed as a bandwidth problem and is addressed by seeking higher-bandwidth solutions.

Today, manufacturers are recognizing this difficulty. They are turning to network processor technologies to manage bandwidth resources more efficiently and to provide the advanced data services, at wire speed, that are commonly found in routers and network application servers. These services include load balancing, QoS, gateways, fire walls, security, and web caching.

For remote access applications, performance, bandwidth-on-demand, security, and authentication rank as top priorities. The demand for integration of QoS and Class of Service (CoS), integrated voice handling, and more sophisticated security solutions will also shape the designs of future remote access network switches. Further, remote access will have to accommodate an increasing number of physical mediums, such as ISDN, T1, E1, OC-3 through OC-48, cable, and xDSL modems.

Industry consultants have defined a network processor (herein also mentioned as an "NP") as a programmable communications integrated circuit capable of performing one or more of the following functions:

Packet classification -- identifying a packet based on known characteristics, such as address or protocol;

Packet modification -- modifying the packet to comply with IP, ATM, or other protocols (for example, updating the time-to-live field in the header for IP);

Queue/policy management -- reflects the design strategy for packet queuing, de-queuing, and scheduling of packets for specific applications; and,

Packet forwarding -- transmission and receipt of data over the switch fabric and forwarding or routing the packet to the appropriate address.

Although this definition is an accurate description of the basic features of early NPs, the full potential capabilities and benefits of NPs are yet to be realized. Network processors can increase bandwidth and solve latency problems in a broad range of applications by allowing networking tasks previously handled in software to be executed in hardware. In addition, NPs can provide speed improvements through architectures, such as parallel distributed processing and pipeline processing designs. These capabilities can enable efficient search engines, increase throughput, and provide rapid execution of complex tasks.

Network processors are expected to become the fundamental network building block for networks in the same fashion that CPUs are for PCs. Typical capabilities offered by an NP are real-time processing, security, store and forward, switch fabric interface, and IP packet handling and learning capabilities. NPs target ISO layer two through five and are designed to optimize network-specific tasks.

The processor-model NP incorporates multiple general purpose processors and specialized logic. Suppliers are turning to this design to provide scalable, flexible solutions that can accommodate change in a timely and cost-effective fashion. A processor-model NP allows distributed processing at lower levels of integration, providing higher throughput, flexibility and control. Programmability can enable easy migration to new protocols and technologies, without requiring new ASIC designs.

Commensurate with design and implementation of network processor-based devices, are the data structures and methods for defining and controlling the overall behavior of an NP, to which this invention is directed.

Summary of the Invention

Accordingly, the invention provides a system for controlling overall behavior of a network processor implemented in a network processing environment servicing a communications network, said system comprising: an interface device located in said network processor for receiving a guided control frame including one or more control functions for configuring various functional devices within said network processor with device control parameter data; means for forwarding said one or more control functions from a received control frame to a functional device within said

network processor to be configured; and processing means implemented in said functional device for executing control functions as specified in the control frame, whereby said device is controlled in accordance with said executed control functions and device control parameter data.

5

According to a preferred embodiment, there is provided a system and method providing functionality for enabling a General Purpose Processor (GPP) that is acting as a Control Point processor (CP) in a network environment to define and control the overall behavior of a Network Processor (NP).

10

Further there is preferably provided a method providing functionality for enabling a General Purpose Processor (GPP) that is acting as a Control Point processor (CP) in a network environment to define and control the overall behavior of a Network Processor (NP) to provide supporting functions to frame-forwarding applications running in an NP.

15

According to a preferred embodiment a novel control frame data structure and communications infrastructure is implemented whereby any network processor device operating in a distributed network processing environment may be controlled in accordance with executed control functions and device control parameter data. Functional units in a network processor device particularly targeted for control preferably include hardware Guided Frame Handler (GFH) device, and frame-forwarding applications controlled by a hardware Guided Table Handler (GTH) device.

20

25

Advantageously, the system and methodology is preferably capable of handling and processing control frames (guided frame flows) in a variety of possible flows through a typical NP system including guided frame flows on a primary blade; guided frame flows on a secondary blade; and, guided frame flows on multiple blades in a network processing environment.

30

In a further aspect, the invention provides a data structure for a control frame to be communicated to a network processor device implemented in a network processing environment, said data structure for controlling overall behavior of said network processor, said data structure including: data for indicating a target location of said network processor within said network processing environment, said target location including an address location of said network processor; data including a target functional device operating within said network processor device; and data including control functions and associated control parameters for

35

40

configuring said target functional device operating in said target network processor, whereby a first network processor receiving said control frame forwards said control frame to said target network processor for configuring said functional device thereof with said control functions and associated control parameters.

In a still further aspect, the invention provides a method for controlling overall behavior of a network processor implemented in a network processing environment servicing a communications network, said method comprising the steps of:

a) receiving a guided control frame including one or more control functions for configuring various functional devices within said network processor with device control parameter data;

b) forwarding said one or more control functions from a received control frame to a functional device within said network processor to be configured; and

c) executing control functions as specified in the control frame to thereby configure said device, whereby said network processor is controlled in accordance with said executed control functions and device control parameter data.

Brief Description of the Drawings

Embodiments of the present invention will now be described by way of example only and with reference to the following examples:

Figure 1 illustrates an example network processing environment implementing one or more network processor devices.

Figure 2 illustrates a block diagram of a Guided Frame Handler component of a network processor device depicted in Figure 1 in accordance with one embodiment of the present invention.

Figure 3 illustrates a block diagram of a Guided Table Handler component of a network processor device depicted in Figure 1 in accordance with one embodiment of the present invention.

Figures 4(a) and 4(b) illustrate example messaging scenarios depicting guided frame flows between a CP and a GFH (Figure 4(a)) on a single blade and, a GDH and a GFH (Figure 4(b)) on a single blade.

Figures 5(a) and 5(b) illustrate example messaging scenarios depicting guided frame flows between a CP and a GFH (Figure 5(a)) on a secondary blade and, a GDH and a GFH (Figure 5(b)) on a secondary blade.

Figures 6(a) and 6(b) illustrate example messaging scenarios depicting guided frame flows between a CP and a GTH (Figure 6(a)) on a single blade and, a CP and a GTH (Figure 6(b)) on a secondary blade.

Figures 7(a) and 7(b) illustrate example messaging scenarios depicting guided frame flows between a CLP and a GTH (Figure 7(a)) on a single blade and, a CLP and a GTH (Figure 7(b)) on a secondary blade.

Figure 8 is a flow diagram 310 depicting the processing of a guided frame in the GFH in accordance with one embodiment of the present invention.

Figure 9 is a flow diagram 350 depicting the guided frame processing of a guided frame in accordance with one embodiment of the present invention.

Figure 10 is a flow diagram 400 depicting the guided command processing of commands within guided frame in accordance with one embodiment of the present invention.

Figure 11 is a flow diagram 450 depicting processing of guided frames that are not unicast and are not destined for present blade in accordance with one embodiment of the present invention.

Figure 12 is a detailed illustration depicting the frame control information included in a guided frame in accordance with one embodiment of the present invention.

Detailed Description of the Preferred Embodiment

Figure 1 depicts generally an example network processing environment implementing one or more network processor ("NP") interface devices 15a,b for servicing a network (not shown). The preferred structure and operation of the network processor interface device 15a,b is as described in commonly-owned, CO-pending U.S. Patent Serial Application 09/384,691 filed August 27, 1999 and entitled "NETWORK PROCESSOR PROCESSING COMPLEX AND METHODS", a copy of which is placed on the file.

With reference to the example NP environment 10 depicted Figure 1, one NP Interface device 15a is provided on a primary blade (i.e., printed circuit board element) 20a with a second NP interface device 15b provided on a secondary blade 20b. It is understood that the system may be expanded to include many blades [primary and secondary(s)] and CP subsystems, for example, in a distributed network configuration. A Control Point subsystem ("CP") 50 includes a system processor 52 that is connected to the NP environment configuration. The system processor at the CP, among other things, preferably provides initialization and configuration services to the NP device by issuing system control frames herein referred to as "guided frames". The CP 50 may be located in any of three locations: in the interface device chip; on the blade on which the chip is mounted (as depicted in Figure 1); or, external to the blade. If external to the blade, the CP may be remote; that is, it is housed elsewhere and communicates via a network connecting the interface device and the CP 50. The elements of a CP generally include a processor 52, memory elements (cache, flash and SDRAM), a memory controller, a PCI bus, and connectors for a backplane and for L1 network media.

As will be described in greater detail herein, in the preferred embodiment, a guided frame is a variable word structure including the following information: 1) frame control information comprising data for indicating which blade the frame is meant for, and to which functional block within the frame it should be sent; 2) correlator information for indicating in the frame where any response frame is to be sent if an error or exception occurs or if frame processing is successful; 3) guided commands including control operands that control or manage the functional units within the blades, e.g., fill a memory location with data; and, 4) an end delimiter command indicating the end of the frame. Each of these will be discussed in greater detail with reference to Figure 12.

As depicted in Figure 1, each NP device includes an embedded processor complex (EPC) 16 having two functional sub-assemblies: an Ingress (up-side) sub-system 26a for receiving/processing data from an external network serviced by the NP system 10 via network protocol-specific data mover interfaces (DM) 24a,b; and, an Egress (down-side) sub-system 26b for transmitting data outbound to a network serviced by the NP system 10 via network protocol-specific data mover interfaces (DM) 24c,d. All frames that are passed from the Ingress to the Egress within the same NP device are passed through Data Align Synchronous Link (DASL) interfaces 45a-45d a detailed description of which may be found in CO-pending U.S. Patent Application Ser. No. 09/330,968 filed 11

June 1999 and entitled "High Speed Parallel/Serial Link for Data Communication" a copy of which is placed on the file. It is understood that other equivalent type interface/link supporting high data flow rates may be implemented. Frames that need to be communicated internally from the egress side 26b back to a processor on the ingress side 26a may be communicated via a data mover wrap (DM_W) 25.

Generally, the ingress sub-assembly 26a of the NP device includes an embedded Guided Frame Handler unit 18 (GFH) and one or more General Data Handlers (GDH) processing units 17a each functioning to classify a received frame and perform a lookup to see where to forward the frame. The egress sub-assembly 26b of the NP device also includes an embedded Guided Frame Handler unit 18d (GFH), one or more Classifier/Look-up processors 17b each functioning to perform any outbound modifications or lookups, if necessary, before transmitting a frame, and additionally, includes a Guided Table Handler unit 19 (GTH). In the NP processor device, all received data traffic is handled by special Classifier/Lookup Processors referred to herein as Guided Data Handlers (GDH), and, as will be described in greater detail herein, control information used to manage the NP system 10 is handled by the Guided Frame Handler (GFH) 18. In the frame forwarding process, the Guided Table Handler (GTH) 19 implements forwarding tables.

With particular reference to Figure 1 and the CO-pending U.S. Patent Application Serial No. 09/384,691, the general flow of a packet or frame received at the NP device is as follows: frames received from an network, e.g., Ethernet MAC, are received at a data mover interface (DM) 24a, 24b where they are identified as either normal data frames or system control frames, (Guided Frames) received from a control processor (CP) system 50. Guided frames are first generally enqueued to an Embedded Processor Complex (EPC) which comprises a plurality of picoprocessors, e.g., protocol processors, which execute logic (picocode) capable of looking at the received frame header and deciding what to do with the frame (forward, modify, filter, etc.). The EPC has access to several lookup tables, and classification hardware assists to allow the picoprocessors to keep up with the high-bandwidth requirements of the Network Processor. A classification hardware assist device is provided for classifying frames of well known frame formats. The Embedded Processing Complex (EPC) provides and controls the programmability of the NP device chip and includes, among other components (such as memory, dispatcher, DM interfaces), one or more processing units which concurrently execute picocode that is stored in a common instruction memory. Each processing

unit preferably includes a CLP Processing Unit core, for example, comprising a 3-stage pipeline, general purpose registers and an ALU. In operation, classification results from the classification hardware assist device are passed to a one or more processing unit during frame dispatch. One processing unit, in particular, referred to as a General Data Handler (GDH) comprises a full CLP processing unit and five coprocessors and are primarily used for forwarding frames. At least two of the N protocol processors are specialized; one for handling Guided Frames -the Guided Frame Handler, GFH 18, and one for building Lookup Data in Control Memory -the Guided Table Handler, GTH 19.

In the preferred embodiment, all frames classified as guided frames are enqueued to an Ingress queue of the GFH on the blade receiving the frame. Figure 2 depicts the general functional layout of the guided frame handler GFH unit 18. A detailed description of the GFH architecture may be found in above-mentioned commonly-owned, CO-pending U.S. Patent Serial Application 09/384,691. Preferably, the GFH 18 is a hardware device that implements software components including a Guided Frame Processor 181 that is responsible for performing guided frame parsing and routing functions; and, a guided command processor 182 that is responsible for interpreting guided "commands" in the guided frame and ensuring that these commands are processed. The Guided Frame Processor 181 receives and processes the guided frames and is responsible for validating the guided request frame and determining whether the GFH is capable of processing the received frame. It also determines the type of acknowledgment (if any) that is required by the sender. The Guided Frame Processor 181 additionally runs through the request frame until it finds the first guided command to be processed and then passes control to the Guided Command Processor 182. On completion of a command, control is passed back to the Guided Frame Processor 181. Otherwise, the Guided Command Processor processes each guided command of the frame until either the Guided Command Processor returns an error or an End_Delimiter Command is encountered. At this point, frame processing is deemed complete.

The Guided Command Processor is responsible for processing a guided command embedded in the guided request frame. It is invoked by the Guided Frame Processor with a reference to the position in the request frame where the command is to be found. The Guided Command Processor parses the command identifier to determine whether it is capable of processing the command. If the specified command is unsupported, an error is returned to the Frame Parser. Else, in the manner described herein, control is

transferred to the appropriate functional block of the blocks 185 shown in Figure 2, which actually services the command.

Figure 3 depicts the functional layout of the GTH unit 19. A detailed description of the GTH architecture may be found in above-mentioned, commonly-owned, CO-pending U.S. Patent Serial Application 09/384,691 and is dispatched when the NP dispatcher sees that it is idle and that a guided request frame is available in the GTH queue. The GTH functions in a similar manner as the GFH in that incoming frames are validated by a guided frame processor 191 and control is passed to a guided command processor 192 for processing of commands. As shown in Figure 3, the service blocks 195 included within the GTH perform different operations than the functional blocks contained in the GFH. It is understood, however, that the purpose of the guided frames are to actually manage and control each of the operational blocks shown in Figures 2 and 3.

What now follows are detailed descriptions of exemplary GFH message flow scenarios for controlling and managing the NP device behavior. In a first example, illustrated in Figures 4(a) and 4(b), there is depicted an example scenario 200 including message flows between a CP subsystem 50 and a Guided Frame Handler (GFH) 18 on a primary blade 20a.

In the example scenario 200 illustrated in Figure 4(a), the Guided "Request" Frame includes embedded guided commands destined for the Ingress GFH of the primary blade. The GFH processes the guided request as follows: In a first step, the guided frame is generated by the CP 50 and is transmitted to the primary blade 20. In this scenario, it is assumed that the frame includes bits that indicate that a response is required. The frame is received by the Ingress-GFH on the primary blade 20a and parsed. The results of parsing indicate that the frame is destined for itself. Therefore, the embedded guided commands are processed by the Ingress-GFH, and particularly are carried out by the guided command processor. Assuming that the processing was successful, the results are stored in the frame and the frame control information is modified to indicate that frame is now a "Response" frame. It is then enqueued on an Ingress Queue (not shown) with a target blade address that corresponds to the primary blade, and is transmitted to an Egress-GFH of the primary blade 20a via the DASL interface 45a,b. Then, the Egress-GFH 18a parses the frame, notes that the frame is a response frame and enqueues it in an appropriate Egress Queue (not shown). This enables the frame to be

transmitted through a DM interface, e.g., DM 24a, through an interface connected to the CP 50 which eventually receives the response frame.

5 In an alternate scenario 200 than that depicted in Figure 4(a), the Guided "Request" Frame includes embedded guided commands destined for the Egress GFH of the primary blade. In this scenario, a guided frame is generated by the CP 50 and transmitted to the primary blade and it is assumed that the frame includes bits that indicate that a response is required. The frame is received by the Ingress-GFH on the primary blade
10 and is parsed. In this case, the results of parsing indicate that the frame is destined for the Egress-GFH. Therefore, the frame is enqueued in the Ingress Queue with the target blade set to the Primary Blade and, is transmitted to an Egress-GFH of the primary blade 20a via the DASL interface 45a,b. The Egress-GFH then parses the frame and determines that
15 it is destined for itself. Therefore, it processes the frame and builds a response frame. After response frame build, it is enqueued in the appropriate Egress queue which then causes the response frame to be sent out through the DM interface to the interface connected to the CP.

20 In the example scenario 210 illustrated in Figure 4(b), the Guided "Request" Frame is generated internally by a General Data Handler (GDH) and includes embedded guided commands that are destined for the Ingress GFH 18 on the primary blade. The GFH processes the guided request as follows: In a first step, an Ingress-GDH on the primary blade builds a
25 guided frame destined for the Ingress-GFH on the same blade and the frame is enqueued to the Ingress-GFH Queue. The frame is received by the Ingress- GFH and, during frame processing it is determined that it is destined for itself. Therefore, the Ingress-GFH processes the frame. Assuming that no errors were encountered during processing, the frame is
30 enqueued to the Ingress Discard Queue after processing as there was no bit indication that a response frame was required. If an error was encountered, and a bit in the frame header indicates that a Negative Acknowledgment is required, a Negative Acknowledgment frame is built and enqueued to the Ingress Queue for dispatch to the Egress-GFH 18a. Once,
35 the response frame is received by the Egress-GFH, it examines the frame header, learns that it is a response frame and so enqueues it to the appropriate Egress Queue for dispatch to the CP 50 where the Negative Acknowledgment frame response frame is received.

40 In a first alternate scenario than scenario 210 depicted in Figure 4(b), the Guided "Request" Frame is generated internally by a General Data Handler (GDH) and includes embedded guided commands that are destined for

the Egress GFH 18a on the primary blade. In this scenario, an Ingress-GDH 20 on the primary blade builds a guided frame destined for the Egress-GFH. Preferably, there are two places where this frame may be built: 1) on the Ingress Data Store; or 2) on the Egress Data Store. If the frame is built on the Ingress Data Store, the Ingress-GDH enqueues the frame to the Ingress GFH Queue. The Ingress-GFH receives and parses the frame. As the frame indicates that it is destined for the Egress-GFH, the Ingress-GFH enqueues the frame to the Ingress Queue with the target blade set to the Primary Blade and, is transmitted to an Egress-GFH of the primary blade 18a via the DASL interface 45c,d. Alternatively, the GDH may directly enqueue the frame to the Ingress Queue with the target blade set to the Primary Blade. If however, the frame is built on the Egress Data Store, the Ingress-GDH enqueues the frame to the Egress GFH Queue so that the frame may be received and processed by the Egress-GFH. Assuming that no errors were encountered, the frame is enqueued to the Egress Discard Queue after processing. If an error was encountered and the frame header included a bit indicating that a Negative Acknowledgment is required, a Negative Acknowledgment frame is built and enqueued to the appropriate Egress interface Queue for eventual receipt by the CP 50 via a DM interface.

In a second alternate scenario than the scenario 210 depicted in Figure 4(b), the Guided "Request" Frame is generated internally by an Egress-GDH 20a on the primary blade and includes embedded guided commands that are destined for the Egress GFH 18a on the primary blade. In this scenario, an Egress-GDH 20a on the primary blade builds a guided request frame using the Egress Data Store, destined for the Egress-GFH on the primary blade. The Egress-GDH thus enqueues the frame to the Egress GFH Queue where it is received and processed by the Egress-GFH. Assuming that the processing was successful, the frame is enqueued to the Egress Discard Queue. However, if an error was encountered and the frame indicated that a Negative Acknowledgment was required, a Negative Acknowledgment frame is built and enqueued to the appropriate Egress interface Queue for eventual receipt by the CP 50 via a DM interface.

In a third alternate scenario than scenario 210 depicted in Figure 4(b), the Guided "Request" Frame is generated internally by an Egress-GDH 20a on the primary blade and includes embedded guided commands that are destined for the Ingress GFH 18 on the primary blade. In this scenario, an Egress-GDH on the primary blade builds a guided frame destined for the Ingress-GFH at either of the Egress Data Store or the Ingress Data Store locations. If the frame is built on the Egress Data Store, the Egress-GDH

enqueues the frame to the Wrap DM Queue 25 which enables the frame to reach the Ingress-GFH. Otherwise, if the frame is built on the Ingress Data Store, the Egress-GDH enqueues the frame to the Ingress GFH Queue for the latter to reach the Ingress-GFH where the frame is received and processed by the Ingress-GFH. Assuming no errors were encountered, the frame is enqueued to the Ingress Discard Queue after processing. If an error was encountered and the frame indicated that a Negative Acknowledgment was required, a Negative Acknowledgment frame is built and enqueued to the Ingress Queue with the target blade set to the Primary Blade for receipt by the Egress-GFH where the frame is parsed. As the frame indicates that it is a response frame, the Egress-GFH enqueues the frame to the appropriate Egress Target Port Queue for eventual receipt by the CP 50 via a DM interface.

In a second example, illustrated in Figures 5(a) and 5(b), there is depicted an example scenario including message flows between a CP subsystem 50 and a Guided Frame Handler (GFH) 18 on a Secondary blade.

In the example scenario 220 illustrated in Figure 5(a), the Guided "Request" Frame includes embedded guided commands destined for an Egress GFH on a secondary blade 20b. The GFH processes the guided request as follows: In a first step, the guided frame is generated by the CP and transmitted to the primary blade. In this example scenario, it is assumed that the frame control information includes bits that indicate that the frame is targeted for the Egress-GFH residing on a secondary blade and that a response is required. The frame is received by the Ingress-GFH 18 on the primary blade 20a and is parsed. As a result of parsing, it is determined that the frame is destined for the Egress GFH 18c on a secondary blade 20b. The frame is thus enqueued to the Ingress Queue with the target blade address set appropriately so that it may be dispatched to the Egress-GFH residing on the targeted blade. Upon receiving the frame, the Egress-GFH 18c parses the frame, and determines that the frame is destined for itself. The Egress-GFH processes the request frame and builds a response frame. The Egress-GFH then sends the response to the Ingress-GFH 18b component on the same blade by enqueueing the response to the Wrap Queue DM 25. The Ingress-GFH frame 18b implements a parsing operation and determines that the frame is a response frame. The frame is thus routed to the CP 50 by enqueueing the frame on the Ingress Queue after setting the target blade address to that of the primary blade 18a. The frame is thus received via the DASL interface 45c by the Egress-GFH 18a on the primary blade where it is parsed. As the frame control information indicates that it is a response frame, the Egress-GFH 18a enqueues the

frame to the appropriate Egress interface Queue. This enables the frame to be transmitted through a DM interface, e.g., DM 24a, through a port connected to the CP 50 which eventually receives the response frame.

5 In an alternate scenario related to example scenario 220 depicted in Figure 5(a), the Guided "Request" Frame includes embedded guided commands destined for the Ingress-GFH 18b. In this scenario, the guided frame is generated by the CP 50 and first transmitted to the primary blade. It is assumed that the frame control information includes bits that indicate
10 that the frame is targeted for the Ingress-GFH 18b residing on a secondary blade and that a response is required. Thus, the frame is received by the Ingress-GFH 18 on the primary blade and parsed, resulting in the determination that the frame is destined for the Ingress GFH on a secondary blade. In response, the frame is enqueued to the Ingress Queue
15 with the target blade address set appropriately so that it may be dispatched to the Egress-GFH residing on the targeted blade. The Egress-GFH parses the frame, determines that the frame is destined for the Ingress component of itself and, in response, enqueues the frame to the Wrap Queue DM 25 so that it may be received by the Ingress-GFH 18b. The
20 Ingress-GFH 18b then processes the request frame and builds a response frame. The response frame is routed back to the CP 50 by enqueueing the frame on the Ingress Queue after setting the target blade address to that of the primary blade where it is received via the DASL interface 45b by the Egress-GFH 18a on the primary blade. The frame is parsed at the
25 Egress-GFH 18a on the primary blade and it is determined that the frame is a response frame. In response, the Egress-GFH enqueues the frame to the appropriate Egress interface Queue. This enables the frame to be transmitted through a DM interface, e.g., DM 24a, through a port connected to the CP 50 which eventually receives the response frame.

30 In the example scenario 240 illustrated in Figure 5(b), the Guided "Request" Frame is generated by the GDH on a secondary blade 20b and includes embedded guided commands destined for an Ingress 18b or Egress GFH 18c on the secondary blade 20b. In this example scenario 240, the
35 flow is identical to that described for the like primary blade example scenario as described herein with respect to Figure 4(b). The only difference is in the path followed by the Negative Acknowledgment message generated by the GFH in case of an error. That is, in the case of the Ingress-GFH processing the request frame, the Negative Acknowledgment
40 frame is enqueued to the Ingress Queue, specifying the primary blade 20a as the target blade. In the case of the Egress-GFH, this message is wrapped to the Ingress-GFH on the secondary blade via the DM_W interface

25. The Ingress-GFH routes the frame to the Egress-GFH of the primary blade by enqueueing it to the Ingress Queue with the target blade set to the Primary Blade. When the Egress-GFH of the primary blade receives the frame, it parses the frame and determines that the frame is a response frame. In response, the Egress-GFH enqueues the frame to the appropriate Egress Target Interface Queue. This enables the frame to be transmitted through a DM interface, e.g., DM 24a, through a port connected to the CP 50 which eventually receives the response frame.

As will now be described with respect to Figures 6(a) and 6(b), there is an example scenario including multi-blade message flows, e.g., between a CP subsystem 50 and a Guided Table Handler (GTH) 19 on both primary and secondary blades. The message flows associated with this example scenario is a combination of the messages flows described for primary and secondary blades. When a frame is received from the CP by the Ingress-GFH on the primary blade 18, it is parsed. The results of parsing indicate that the frame is destined for GTHs on multiple blades. The frame is therefore forwarded to all the targeted blades by enqueueing the frame to the Ingress Queue with the list of target blades. The processing of the frame when it is received by the Egress-GFH on each target blade remains the same as described previously. It should be noted that multiple response frames will be sent to the CP corresponding to the number of blades that were originally targeted. For the related example scenarios where the guided frames are generated by a GDH and destined for GTHs on primary and secondary blades, this is similar to the case described above with the condition that no responses are ever sent back to the originating GDH. If Negative Acknowledgment is specified then, under error conditions, one or more Negative Acknowledgment frames will be sent to the CP.

Thus, in the third example scenario 260 illustrated in Figure 6(a), the Guided "Request" Frame is generated by the CP 50 and includes embedded guided commands destined for a GTH 19 on the primary blade 20a. For this example scenario, it is assumed that the frame control information includes bits that indicate that the frame is targeted for the GTH residing on the egress of the primary blade and, that a response is required. In operation, the frame is received by the Ingress-GFH 18 on the primary blade and is parsed. The results of parsing indicate that the frame is destined for the GTH 19 on the primary blade 20a. The frame is thus enqueued on the Ingress Queue with a target blade address that corresponds to the primary blade. This causes the frame to reach the Egress-GFH 18a of the primary blade via the DASL interface 45a,b. The

Egress-GFH 18a parses the frame, notes that the frame is destined for the GTH and enqueues the frame on the Egress-GTH Queue for dispatch to the GTH on the primary blade. The GTH parses the received frame. The results of parsing indicate that a response is required and that the source of the request was the CP. As will be described in greater detail herein with respect to Figure 12, the "Correlator" field of the frame indicates that the source of the frame was the CP 50. The GTH processes the request frame and builds a response frame. The GTH 19 then sends the response frame back to the CP 50 by enqueueing the frame on the appropriate Egress interface Queue. This enables the frame to be transmitted through a DM interface, e.g., DM 24a, through a port connected to the CP 50 which eventually receives the response frame.

In an alternate scenario 280 depicted in Figure 6(b), the Guided "Request" Frame is generated by the CP 50 and includes embedded guided commands destined for the GTH 19 on a secondary blade 20b. Thus, in this scenario it is assumed that the frame contains bits that indicate that the frame is targeted for the GTH residing on a secondary blade and that a response is required. In accordance with the previous described flows, the guided frame is first received by the ingress GFH on the primary blade and parsed. As a result of parsing, it is determined that the frame is destined for the GTH on a secondary blade 20b. The frame is thus enqueue to the Ingress Queue with the target blade address set appropriately for dispatch to the Egress-GFH 18c residing on the targeted blade 20b. The Egress-GFH parses the frame, notes that the frame is destined for the GTH and so enqueues the frame to the Egress GTH Queue. This causes the frame to be received by the GTH which parses the received frame. The results of parsing will determine that a response is required and, it is determined from the "Correlator" field of the frame control information (Figure 12) that the source of the frame was the CP 50. The GTH then processes the request frame, builds a response frame, and sends the response to the Ingress-GFH 18b on the secondary blade via the DM_W interface 25. The Ingress-GFH then performs a parsing operation and determines that this is a response frame. Therefore, the frame is routed to the CP 50 by enqueueing the frame on the Ingress queue after setting the target blade address to that of the primary blade 20a. The frame is received via the DASL interface 45b by the Egress-GFH on the primary blade 20a and is parsed. As the frame indicates that it is a response frame, the Egress-GFH enqueues the frame to the appropriate Egress interface Queue. This enables the frame to be transmitted through a DM interface, e.g., DM 24a, through a port connected to the CP 50 which eventually receives the response frame.

As will now be described with respect to Figures 7(a) and 7(b), there is an example scenario including message flows between a GDH 17a and a GTH 19 on both primary and secondary blades.

Thus, in the fourth example scenario 290 illustrated in Figure 7(a), the Guided "Request" Frame is generated by the Egress - GDH 17a and includes embedded guided commands destined for a GTH 19 on the same (primary) blade. The frame is first routed to the Egress-GFH on the primary blade 20a by enqueueing the frame on the Ingress Queue for dispatch to and receipt by the Egress-GFH 18a via the DASL interface 45a,b. The Egress-GFH 18a parses the frame, notices that it is destined for the GTH 19, and therefore enqueuees the frame on the Egress GTH Queue. This causes the GTH to receive the frame. The GTH then parses the frame and from the Correlator field of the frame control information, determines that the originator of the frame is a GDH. The GTH then processes the frame, and after processing, the frame is discarded by enqueueing it to the Egress Discard Queue. Consequently, no response is sent to the GDH.

In an alternate scenario 300 depicted in Figure 7(b), the Guided "Request" Frame is generated by the GDH 17b and includes embedded guided commands destined for the GTH 19 on a secondary blade 20b. In this scenario, the flow is identical to that described for the primary blade with the only difference being the path followed by the Negative Acknowledgment message generated by the GTH 19 in case of an error. This message is wrapped to the Ingress-GFH 18b on the secondary blade via the DM_W interface 25. The Ingress-GFH routes the frame to the Egress-GFH 18a of the primary blade 20a by enqueueing it to the Ingress Queue where it is eventually forwarded to the CP 50.

Figure 12 depicts the format of the frame control information 60 of a guided frame. As shown in Figure 12, this frame control information 60 includes a two word structure each word 61a, 61b of four (4) bytes each.

In the first word 61a there is included a first byte of data 63 which is a number, e.g., 0CH, that identifies the frame as a control frame. The second byte 65 includes bit flags for providing various processing/routing indications including: a first resp/req flag 70 which is set to indicate whether the frame is a request frame or a response frame; a second ack/noack flag 71 which is set to indicate whether to always reply to the source, e.g., CP, or never reply; a third use/learn flag 72 which is set to indicate whether the blade is to learn where the frame came from (the source address and port of the frame), or whether the source address information is explicitly provided in the frame; a fourth neg/all flag 73

which operates in conjunction with the ack/noack flag 71 for indicating whether a response is to be provided only if an error occurs while processing the frame (neg is set), or if the source always wants a response back regardless if a processing error occurs; a fifth
5 ingress/egress flag 74 which is set to indicate which side of the blade the frame is destined for, i.e., the ingress or egress side; a sixth GTH/GFH flag 75 which indicates what picoprocessor within the blade is to process the flag; and, a seventh ind/chnd flag 76 which is an independent/chained command processing indicator set to indicate that the
10 target preprocessor is to continue processing of guided frame commands even if an error occurs, or, is to stop processing if an error occurs. The next field 66 is a 16 bit field that includes the address of the target blade to which the frame is to be sent. It is understood that this may include unicast or multicast addresses as well. That is, as part of
15 frame processing, the Guided Frame Processor will compare the blade address of the target blade field 66 with the one programmed into the blade as a part of initialization. If these addresses match, then it is assumed that the frame is for that blade only. If these addresses do not match, it is assumed to be targeted for another blade or blades in the
20 system (i.e., is a unicast or multicast address) and, as a result, will be passed up to a switch fabric (not shown) which has the intelligence to determine the blade(s) to which the frame should be targeted.

In the second word 61b of the frame control information 60, there is
25 included at least a first byte of data 67 for indicating the source port from which the guided frame originated. In the scenarios described, this source port may include a data mover (DM) interface port, for example. Two additional bytes of data 68 include a source blade address for the origination frame. It is understood that how these frames are built
30 depends on the use/learn flag 72. For example, the blade will not fill the source port and source blade address fields if the "learn" flag 72 is not set. Thus, the blade processing is responsible for learning where the frame came from. Generally, guided frames that are generated internally (e.g., by a GDH) will not require response frames, i.e., the ack/noack
35 field 71 is set to noack. Thus, these fields will not be set. However, if exceptions/error are to be reported to CP as unsolicited response frames, for example, the NP will be configured to know the source blade and port of the CP.

40 With further regard to the composition of the guided frame, the correlator information that is provided may range anywhere from two to five bytes (not shown) and provides a unique identifier so that when a

response frame is returned to the CP for example, it may be correlated with the original command of the guided request frame. This is preferable because in the course of processing a single guide frame with many embedded commands, multiple response frames may be returned to the requesting CP and each response needs to be correlated.

With regard to the remaining composition of the guided frame, there is provided one or more command fields containing the directives or operations that are used to configure and control the NP devices in an NP networking environment. Preferably, each command field includes at least three (3) sub-fields including: 1) a command control word for identifying the type of command that is to be performed; 2) an address field indicating the address locations within the NP device on which the command(s) is to operate; and, 3) a field of operands of variable length which comprise the data associated with the command type. For example, a command type may include, setup register, or, fill memory location with data and the address information sub-field will provide the address of the register/memory location at which the operand data is to be filled. It is understood that there are many command types which may be implemented for controlling and configuring the functional blocks indicated in the GFH and GTH structures of Figures 2 and 3, respectively. A unique end of frame delimiter is the remaining word in the guided frame to indicate that no more commands are provided and that guided frame processing is to cease.

The previously described guided frame message flow scenarios depicted in the figures are embodied by the flow charts in Figures 8-11 as will now be described.

Figure 8 is a flow diagram 310 depicting the processing of a guided frame in the GFH. As shown in Figure 8, at step 312, the incoming guided frame is received by the GFH. At step 314, the frame is processed by the Guided Frame Processor of the GFH as depicted in Figure 9, where the frame control information is parsed to determine whether it is a valid frame, i.e., includes the guided frame identifier. At step 316, a decision is made as to whether it is a valid frame and if it is not a valid frame, i.e., not a guided control frame, the frame will be discarded at step 318 and the processor will return to an idle state at step 320. If at step 316 it is determined that the frame is a valid guided frame, a decision is made at step 322 to determine whether the received frame is a response frame or a request frame. If it is determined that the frame is a response frame, the frame is processed by the Guided Frame Processor of the GFH at step 324 and proceeds in accordance with the Guided Frame

Processor flow depicted in Figure 9. Otherwise, if the received guided frame is a request frame, then the target blade address is determined at step 326. Then, at step 328, a decision is made as to whether the target blade is the current blade that received the frame. If the target blade in the frame is not destined for the present blade, then the frame is processed at step 330 in accordance with the Alternative Processing flow as depicted in Figure 11. Otherwise, if the received guided frame is destined for the present blade, then a determination is made at step 333 as to whether the up(ingress)/down(egress) flag bit 74 (Figure 12) is set in the first frame control word to indicate that the frame is destined for processing on the egress or ingress side of the blade. If the frame is destined for processing on the egress side, then a further determination is made at step 336 as to whether the frame came from the egress or ingress side of the blade. Likewise, at step 338, if the frame is destined for processing on the ingress side, then a further determination is made to whether the frame came from the egress or ingress side of the blade. If, at step 336, the frame is destined for the egress side and originated at the egress side, then the frame is processed at step 340 in accordance with the Guided Command Processor flow as depicted in Figure 10. Likewise, at step 338, if the frame is destined for the ingress side and originated at the ingress side, then the frame is processed at step 342 in accordance with the Guided Command Processor flow as depicted in Figure 10. If the frame is destined to a side opposite from the side from which it was received, then the frame is routed to the correct side, as indicated at step 345.

Figure 9 is a flow diagram 350 depicting the guided frame processing of a guided frame. As shown in Figure 9, at step 352, an incoming guided "response" frame is received from the GFH. At step 354, a determination is then made as to whether the ack/noack flag bit 71 (Figure 12) is set in the first frame control word to indicate whether the response is to be sent or discarded. If no acknowledgment is required, then the frame will be discarded at step 356 and the processor will return to an idle state at step 358. If at step 354, it is determined that the response frame is to be sent, then a determination is made at step 360 as to whether the neg/all flag bit 73 (Figure 12) is set in the first frame control word to indicate whether a response is to be always sent or whether a response is to be sent only when an error occurs. If the response is always to be sent, then a determination is made at step 362 as to whether the CP is attached locally or is remote. If the CP is local, then the response frame is dispatched to the local CP at step 365. If at step 360, it is determined that a response is to be sent only when an error occurs, then a

determination is made at step 366 as to whether an error actually occurred during the processing. If so then the process proceeds to step 362 for dispatch to the CP whether remote or local. If it is determined that an error did not occur at step 366, then the frame will be discarded at step 368 and the processor will return to an idle state at step 370. Returning to step 362, if the CP to which a response frame is to be set is remote, then a decision is made at step 373 to determine whether the up(ingress)/down(egress) flag bit 74 (Figure 12) is set in the first frame control word indicating where the frame was processed. If it is determined that the frame is at the ingress side, then at step 375, the response frame is forwarded to the DM interface for dispatch to the CP. Otherwise, if it is determined that the frame is at the egress side, then the response frame is wrapped around to the ingress side via the DM_W (wrap) interface at step 378, and the ingress- GFH processes the frame at step 380 in accordance with the incoming loop flow depicted in Figure 8.

Figure 10 is a flow diagram 400 depicting the guided command processing of commands within guided frame. As shown in Figure 10, at step 402, a first determination is made as to whether the GFH/GTH flag bit 75 (Figure 12) is set in the first frame control word indicating where the guided commands are to be processed. At step 404, if the bit flag 75 indicates GFH processing, then at step 406 GFH processing proceeds by retrieving the next command from the guided frame. A determination is then made at step 408 as to whether an end of frame delimiter is encountered, indicating end of guided frame processing. If the end of frame delimiter is encountered, then at step 410 control is returned to the guided frame processor for processing in accordance with the guided frame processor flow depicted in Figure 9. If, at step 408, an end of frame delimiter was not encountered, then at step 415, the picoprocessor registers to be configured are loaded with the required parameters in accordance with the current guided command operands. The call is made to the GFH function (e.g., via a procedure call application program interface) at step 418 to perform the configuration setup, and, at step 420, the function returns control to the GFH guided command processor at step 406 for processing of the next command from the frame. Returning to step 404, if it is determined that the bit flag 75 indicates GTH processing, then at step 409 GTH processing proceeds by retrieving the next command from the guided frame. A determination is then made at step 411 as to whether an end of frame delimiter is encountered, indicating end of guided frame processing. If the end of frame delimiter is encountered, then at step 413 control is returned to the guided frame processor for processing in accordance with the guided frame processor flow depicted in

Figure 9. If, at step 411, an end of frame delimiter was not encountered, then at step 416 the picoprocessor registers to be configured are input with the required parameters in accordance with the current guided command operands. The call is made to the GTH function at step 419 to perform the configuration setup, and, at step 422, the function returns control to the GTH guided command processor at step 409 for processing of the next command from the frame.

As mentioned with respect to step 330, Figure 8, if a request frame received at a primary blade is destined for a secondary or multiple other blades, process proceeds to the Alternative Processing Flow 450 as depicted in Figure 11 at step 452. Specifically, Figure 11 depicts processing 450 of guided frames that are not unicast and are not destined for present blade. Thus, the processing depicted in Figure 11 is directed to frames categorized as follows: 1) unicast frames targeted for another blade; 2) multicast frames targeted for other blades but not the present blade; and, 3) multicast frames targeted for other blades including the present blade. As shown in Figure 11, at step 454, a determination is made as to which of the three categories the current frame applies. If it is determined that the frame is of category 1) and 2), i.e., a unicast frame targeted for another blade or a multicast frame targeted for another blade but not the present blade, then processing proceeds to step 456 where the frame is forwarded to a network processor switching fabric for dispatch to the destined blades on their egress side. Then, as indicated at step 458, the egress GFH of each blade processes the frame in accordance with the incoming loop flow depicted in Figure 8. If at step 454, it is determined that the frame is of category 3), i.e., a multicast frame targeted for other blades including the present blade, then processing proceeds to step 460 where the frame is forwarded to a network processor switching fabric for dispatch to the destined blades including on their egress side, and including the egress side of the present blade. Then, as indicated at step 462, the egress GFH of each blade processes the frame in accordance with the incoming loop flow depicted in Figure 9.

CLAIMS

1. A system for controlling overall behavior of a network processor implemented in a network processing environment servicing a communications network, said system comprising:

an interface device located in said network processor for receiving a guided control frame including one or more control functions for configuring various functional devices within said network processor with device control parameter data;

means for forwarding said one or more control functions from a received control frame to a functional device within said network processor to be configured; and

processing means implemented in said functional device for executing control functions as specified in the control frame, whereby said device is controlled in accordance with said executed control functions and device control parameter data.

2. The system as claimed in Claim 1, further including a control point subsystem in said network processing environment, said subsystem generating said guided control frame.

3. The system as claimed in Claim 1 or 2, further including a processing mechanism for parsing a received frame and determining from said frame to which device said control frame is to be routed.

4. The system as claimed in Claim 2, wherein said forwarding means includes one or more data mover interfaces for receiving control frames from said control point subsystem destined for said network processor and forwarding response frames generated by a device to be controlled back to said control point subsystem.

5. The system as claimed in any preceding claim, wherein said network processor includes an ingress side and egress side each capable of processing frames, said forwarding means including high speed links for forwarding frames between said ingress side and egress sides.

6. The system as claimed in Claim 1, wherein said network processor device includes control processor for generating guided control frames internal to said network processor, said forwarding means forwarding said

control frames to said devices to be configured within said network processor.

5 7. The system as claimed in Claim 5, wherein said network processor includes a wrap data mover interface for forwarding frames from an egress side to said ingress side within said network processor.

10 8. The system as claimed in any preceding claim, wherein said network processing environment includes at least two network processors, a control frame received at a first network processor and including indication control function and device parameter data for controlling devices at another network processor, said forwarding means including means for communicating said control frame to said other network processor from said first network processor.

15 9. The system as claimed in Claim 8, wherein said means for communicating said control frame to said other network processor from said first network processor includes high speed data links.

20 10. The system as claimed in Claim 4, further including mechanism for learning a location of said control point subsystem in said networking environment in an event a response frame is to be returned to said control point subsystem.

25 11. A data structure for a control frame to be communicated to a network processor device implemented in a network processing environment, said data structure for controlling overall behavior of said network processor, said data structure including:

30 data for indicating a target location of said network processor within said network processing environment, said target location including an address location of said network processor;

35 data including a target functional device operating within said network processor device; and

40 data including control functions and associated control parameters for configuring said target functional device operating in said target network processor, whereby a first network processor receiving said control frame forwards said control frame to said target network processor for configuring said functional device thereof with said control functions and associated control parameters.

12. The data structure as claimed in Claim 11, further comprising:

data for indicating whether a response control frame is required to be built when an error occurs during configuration of a functional device; and

data for indicating whether said frame is a request control frame or a response control frame.

13. The data structure as claimed in Claim 11 or 12, further comprising data for indicating whether said control frame is destined to an ingress or egress side of said network processor.

14. The data structure as claimed in Claim 12, wherein said network processing environment includes a control point system for generating said control frames, said control frame further comprising data for indicating a source device to which response frames are to be sent.

15. The data structure as claimed in Claim 14, wherein said control frame further comprises data for indicating whether a functional device to be configured from said control frame is to learn an address location of said control point system sourcing said control frame.

16. The data structure as claimed in Claim 14 or 15, wherein said control functions and associated control parameters for configuring said target functional device are user configurable.

17. A method for controlling overall behavior of a network processor implemented in a network processing environment servicing a communications network, said method comprising the steps of:

a) receiving a guided control frame including one or more control functions for configuring various functional devices within said network processor with device control parameter data;

b) forwarding said one or more control functions from a received control frame to a functional device within said network processor to be configured; and

c) executing control functions as specified in the control frame to thereby configure said device, whereby said network processor is

controlled in accordance with said executed control functions and device control parameter data.

18. The method as claimed in Claim 17, wherein said network processing environment includes a control point subsystem for generating said guided control frame.

19. The method as claimed in Claim 17 or 18, further including the step of parsing a received frame and determining from said frame to which functional device said control frame is to be routed.

20. The method as claimed in Claim 18, wherein said network processor device implements one or more data mover interfaces, said receiving step including receiving control frames from said control point subsystem destined for said network processor via a data mover interface.

21. The method as claimed in Claim 20, wherein during the step of or in response to executing control functions as specified in the control frame, the steps of:

building a response frame in said functional device within said network processor; and

forwarding said response frame back to said control point subsystem.

22. The method as claimed in any of Claims 17 to 21, wherein said network processor includes an ingress side and egress side each capable of processing received frames, said forwarding step further including the step of forwarding frames between said ingress side and egress sides via high speed data links.

23. The method as claimed in Claim 17, wherein said network processor device includes control processor device for generating said guided control frame, said forwarding step including the step of enqueueing said control frames for transmission to said devices to be configured within said network processor.

24. The method as claimed in Claim 22, wherein said network processor includes a wrap data mover interface, said forwarding step further including the step of wrapping frames from said egress side to said ingress side via said wrap interface.

25. The method as claimed in any of Claims 17 to 24, wherein said network processing environment includes at least two network processors, a control frame being received at a first network processor and including indication control function and device parameter data for controlling devices at another network processor, said forwarding step further including the step of communicating said control frame to said other network processor from said first network processor.

26. The method as claimed in Claim 25, wherein said communicating step is conducted via high speed data links.

27. The method as claimed in Claim 21, wherein said step of building a response frame in said functional device further includes the step of learning a location of said control point subsystem in said networking environment in an event a response frame is to be returned to said control point subsystem.

1 / 9

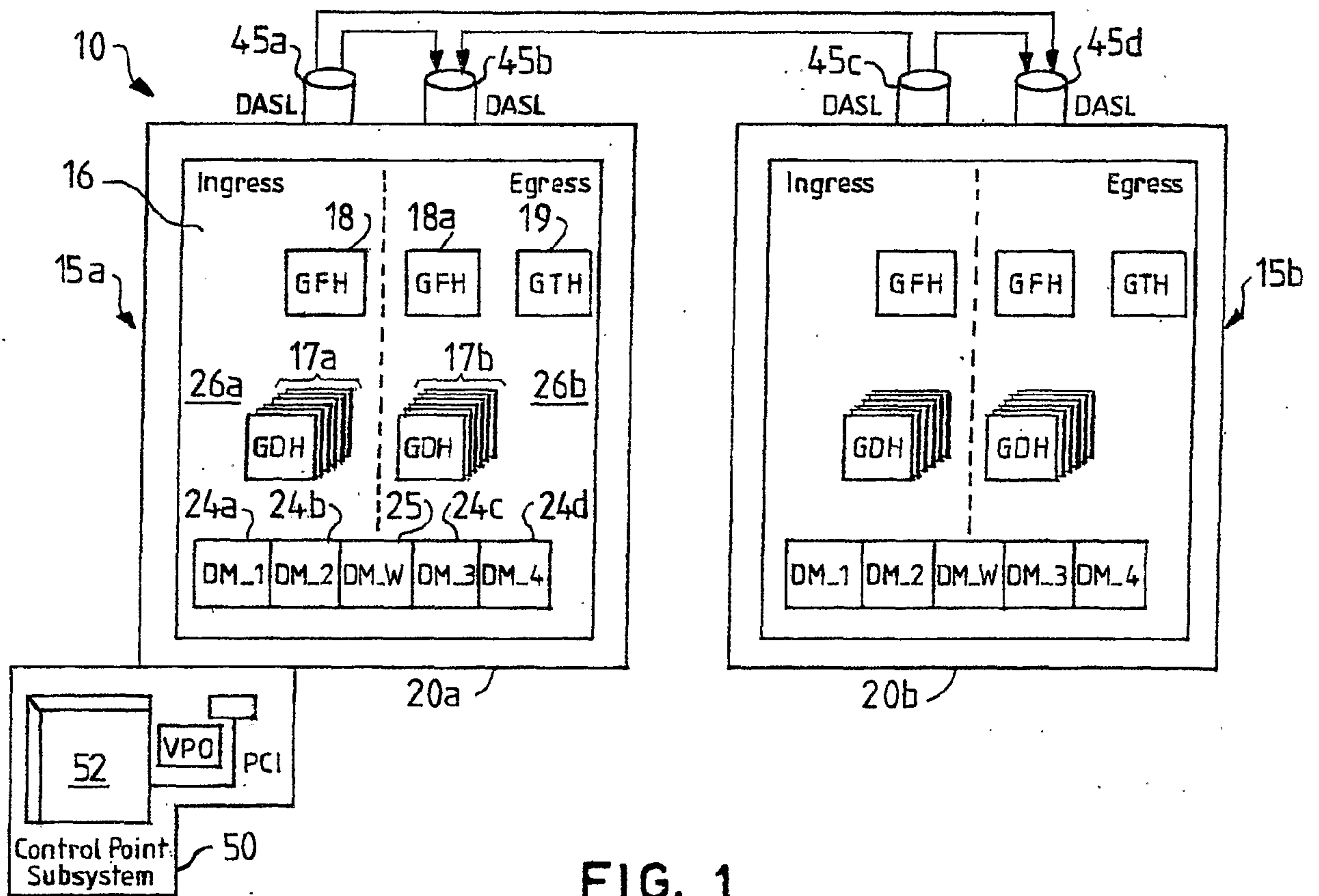


FIG. 1

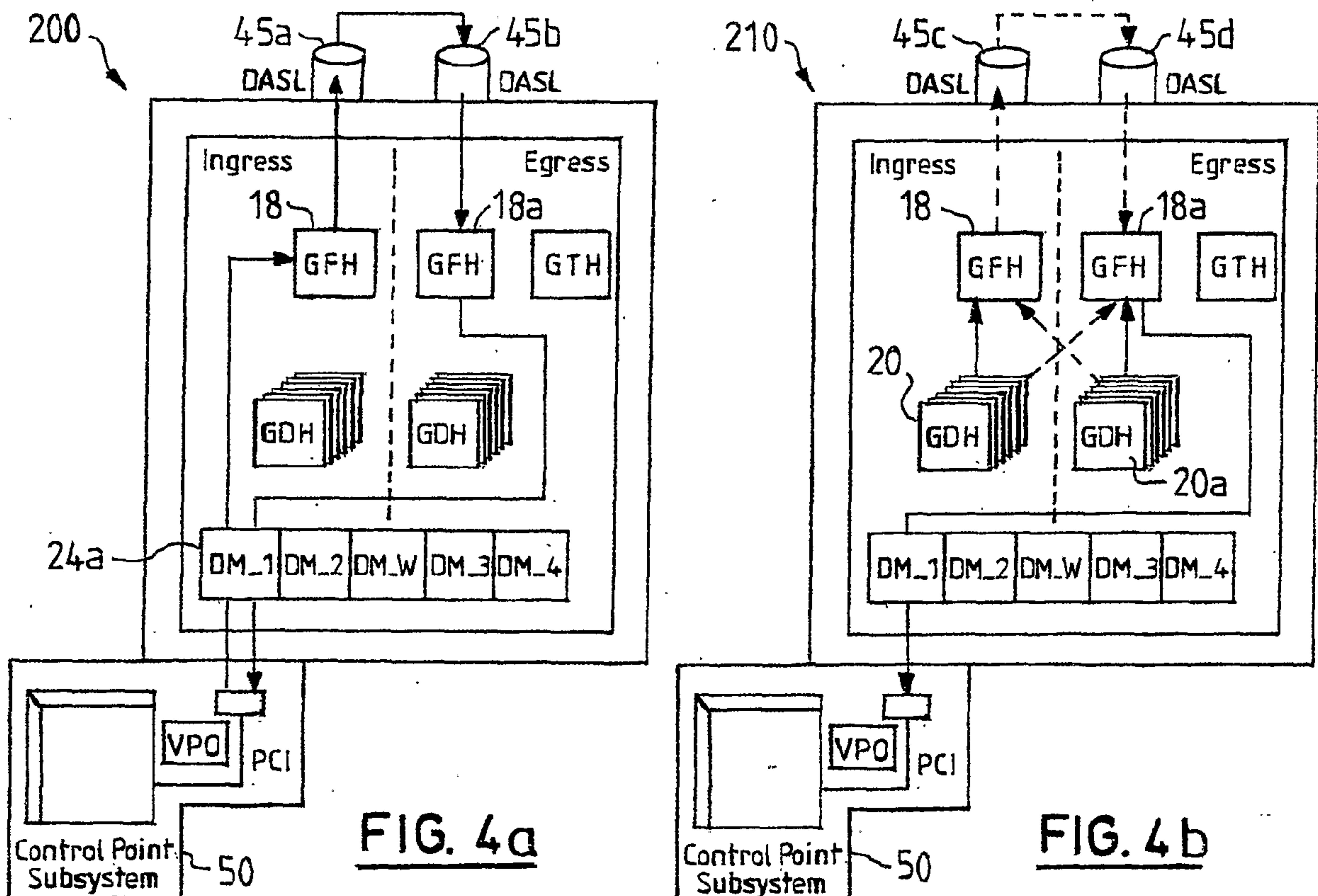
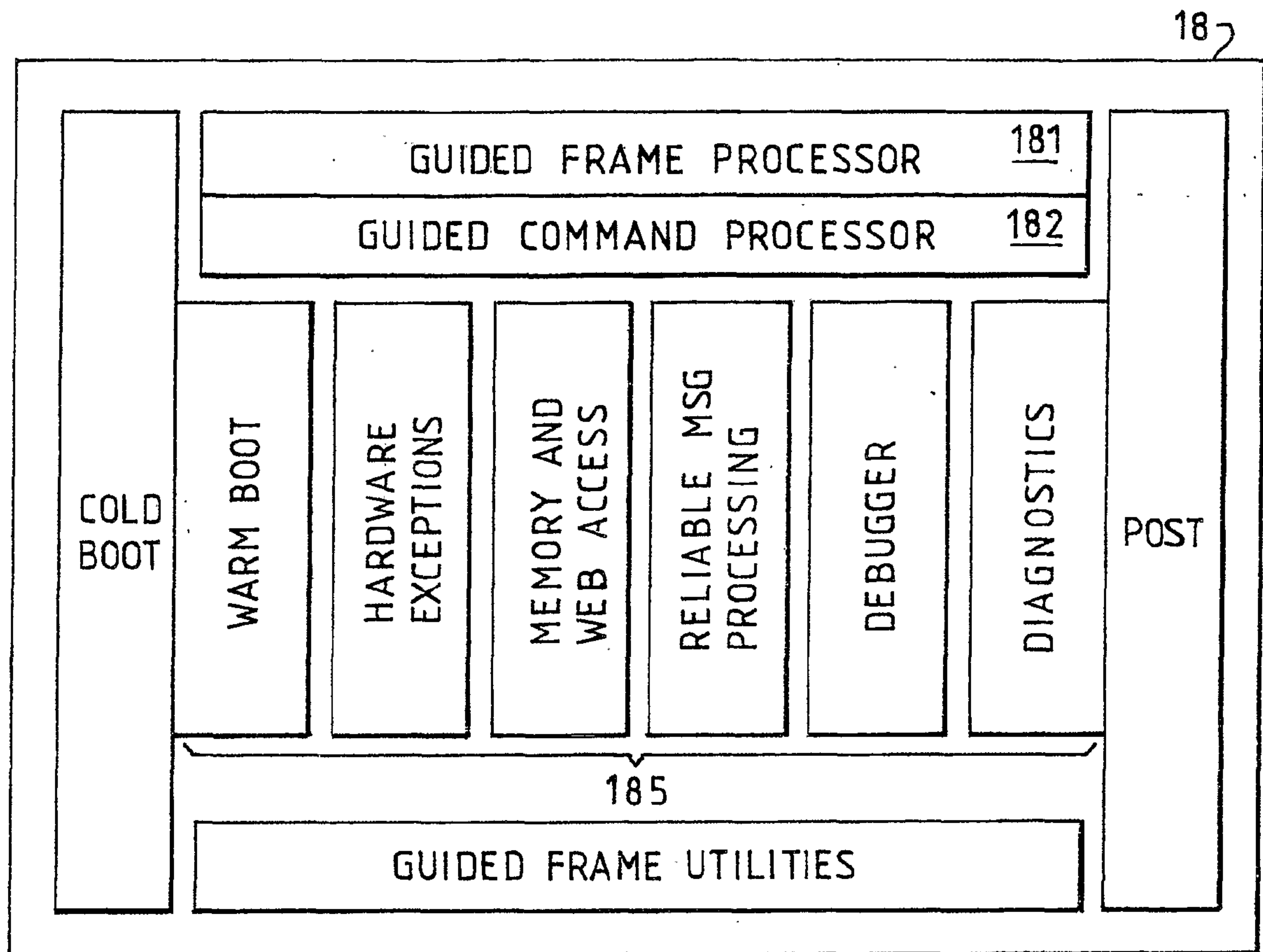
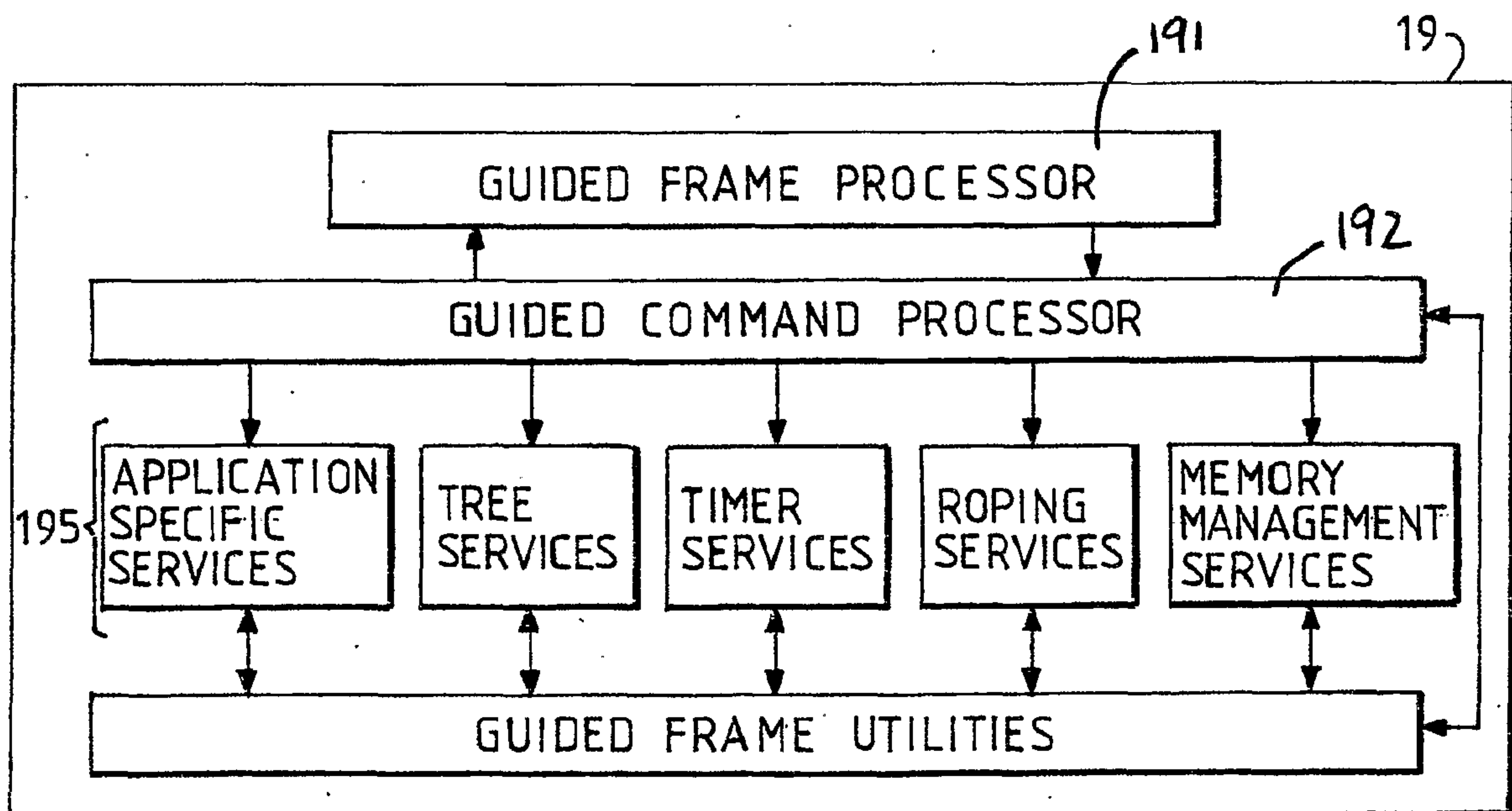


FIG. 4a

FIG. 4b

2 / 9

FIG. 2FIG. 3

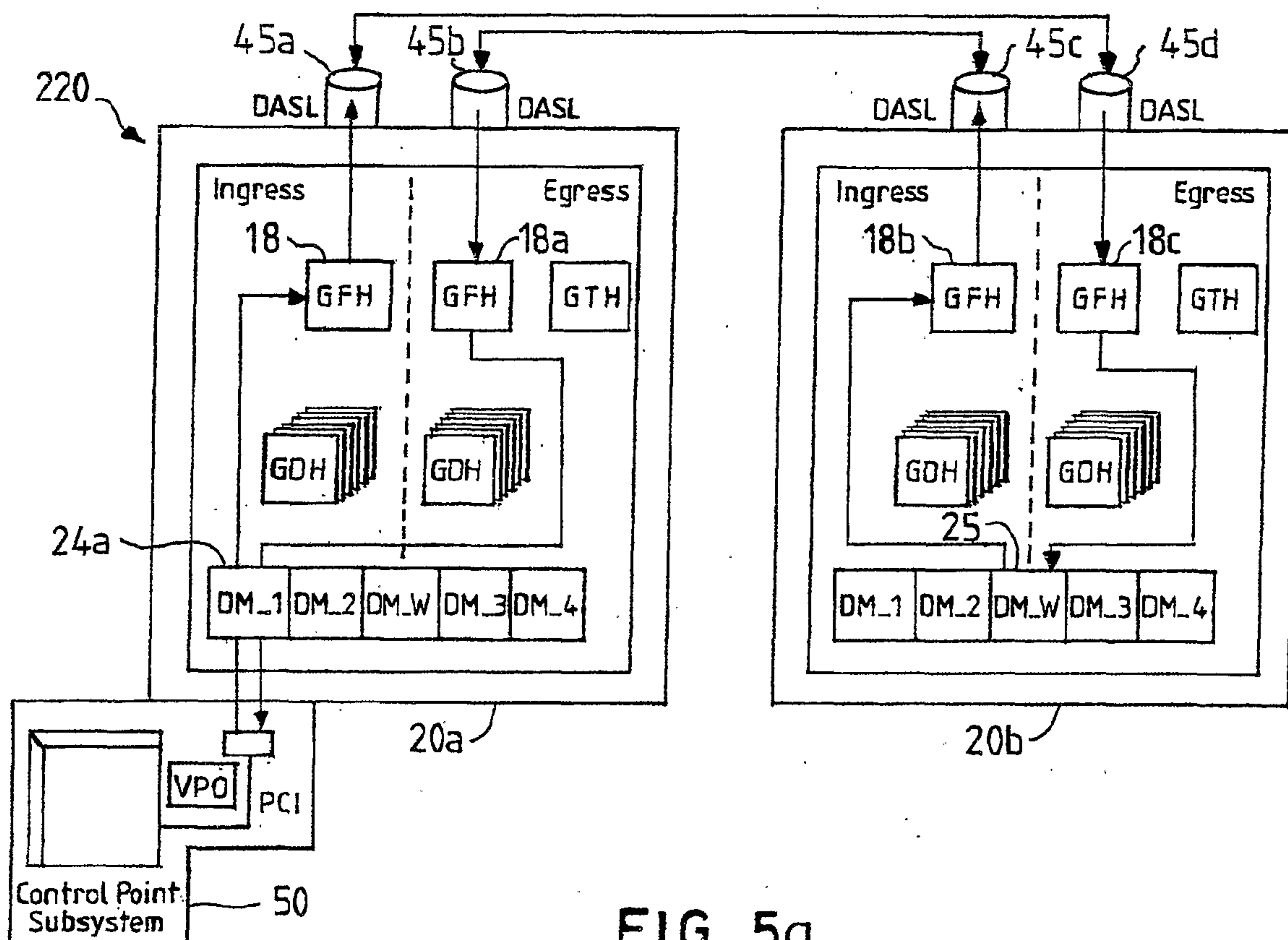


FIG. 5a

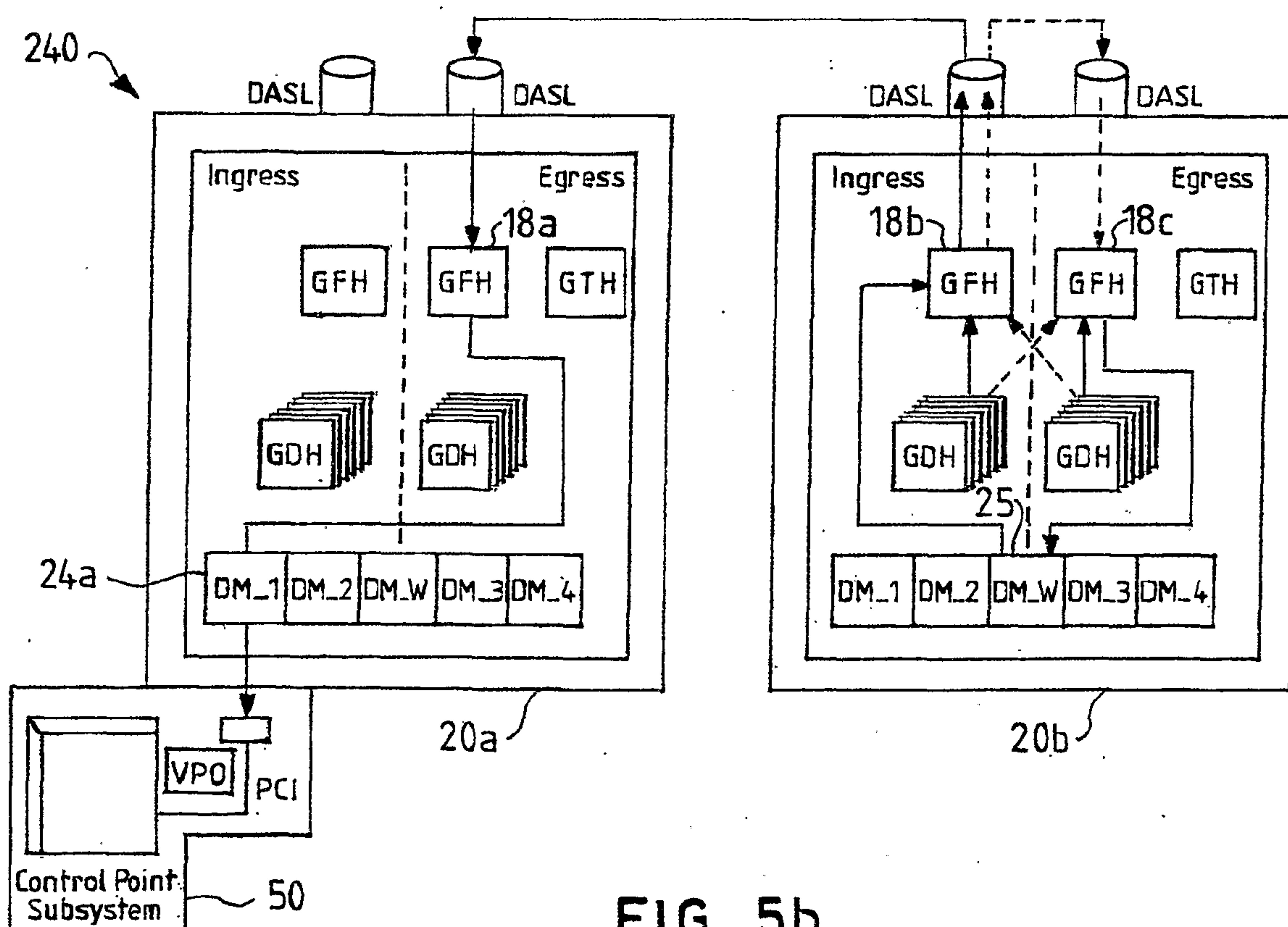
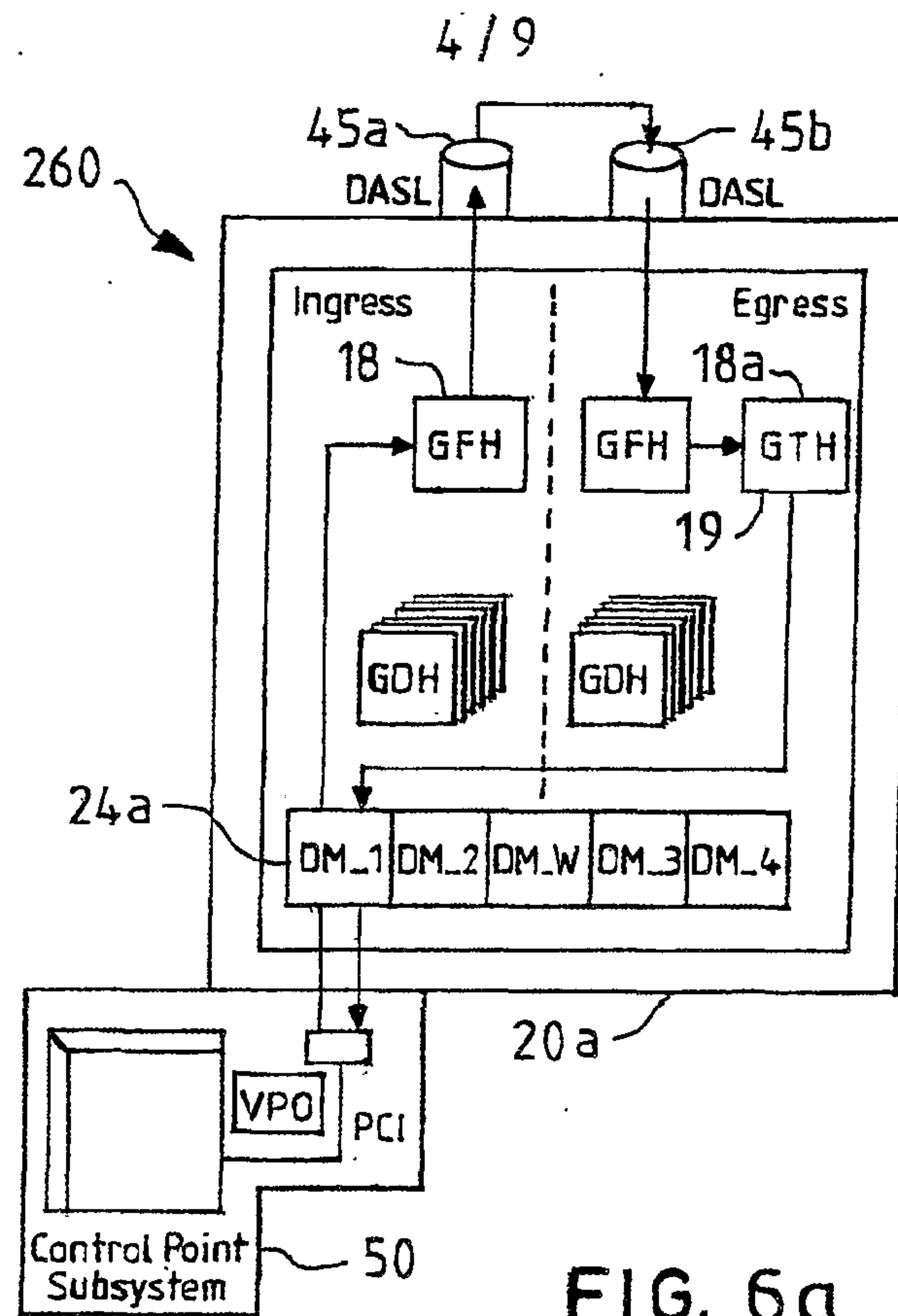
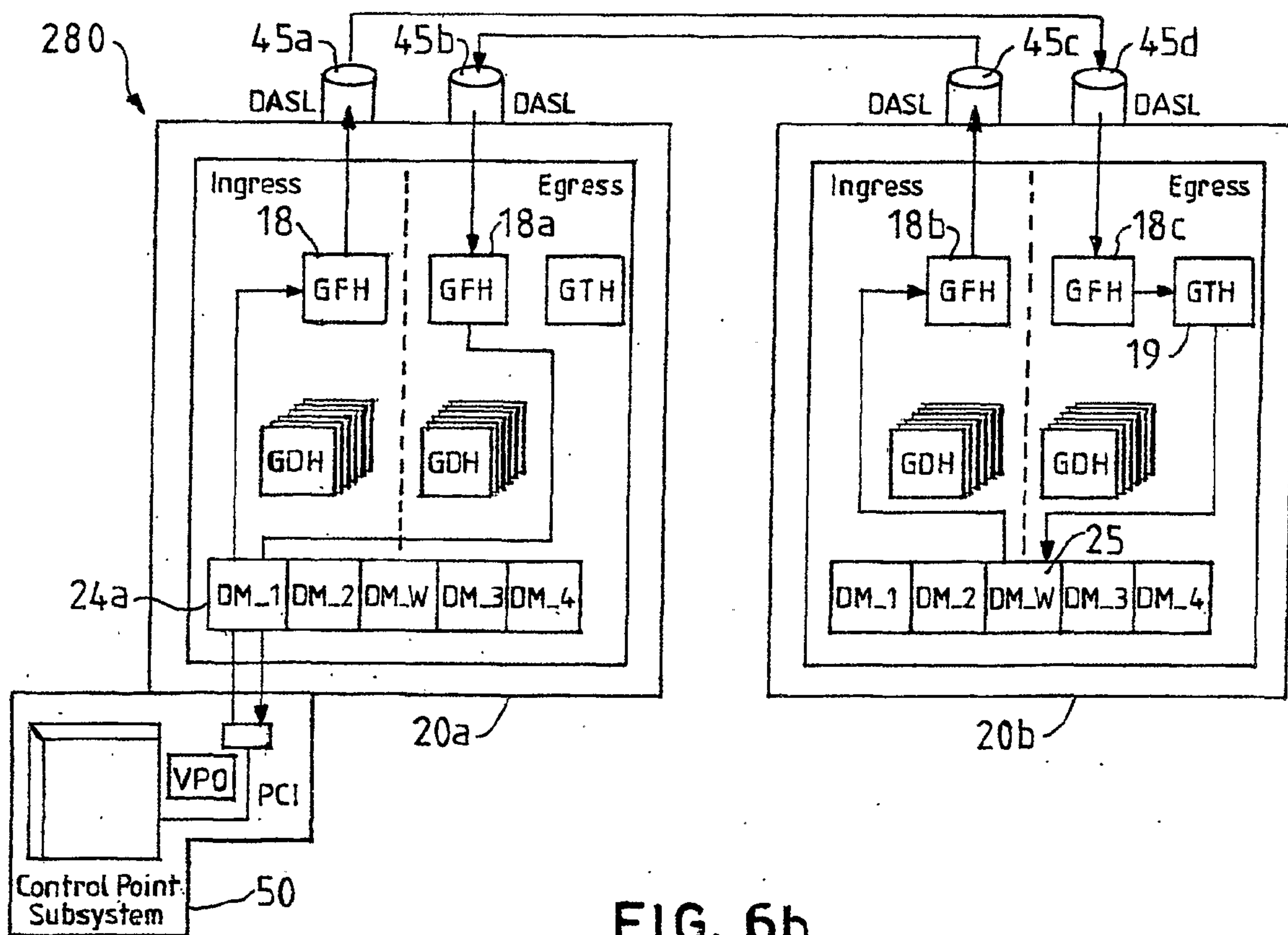
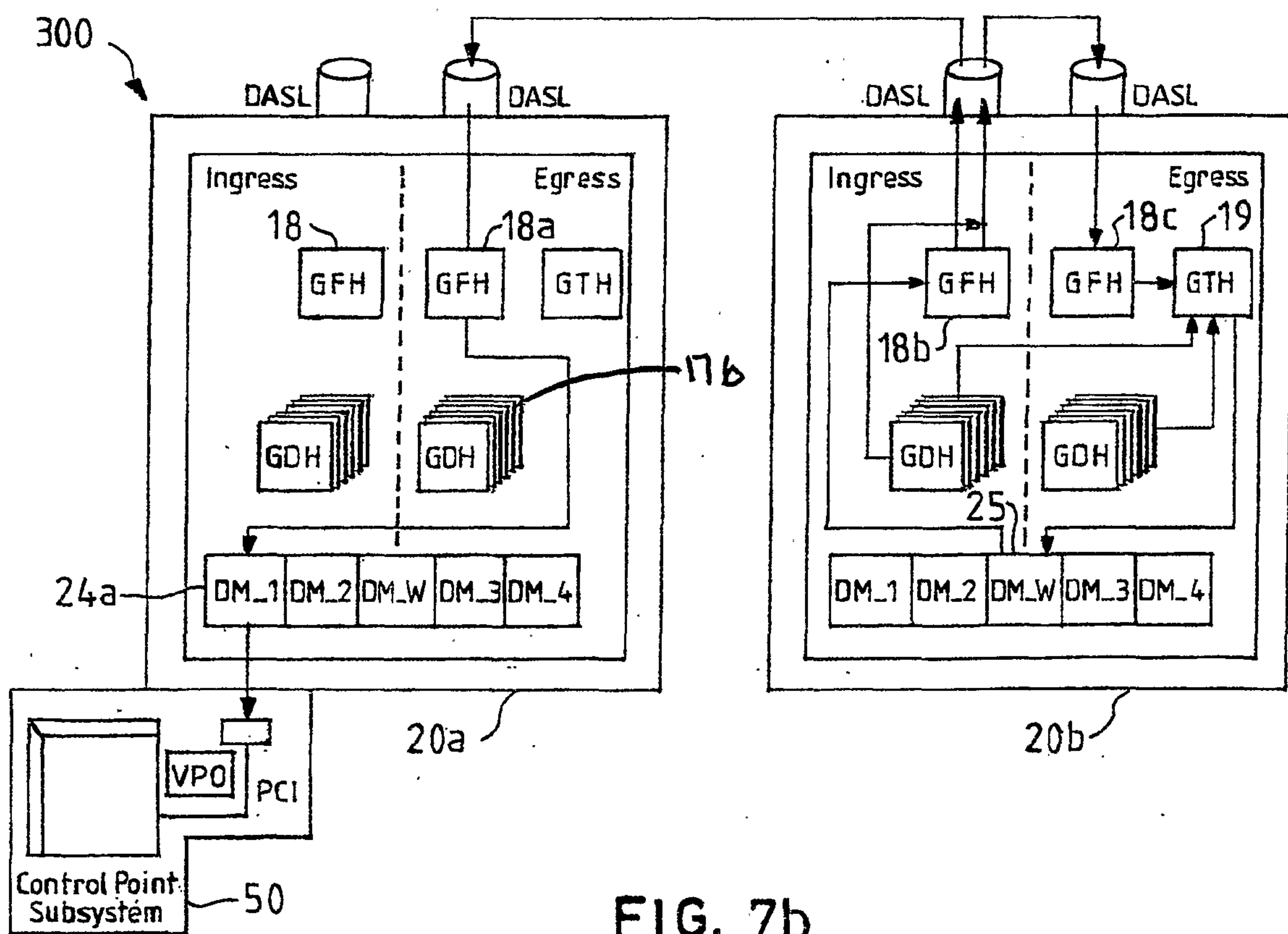
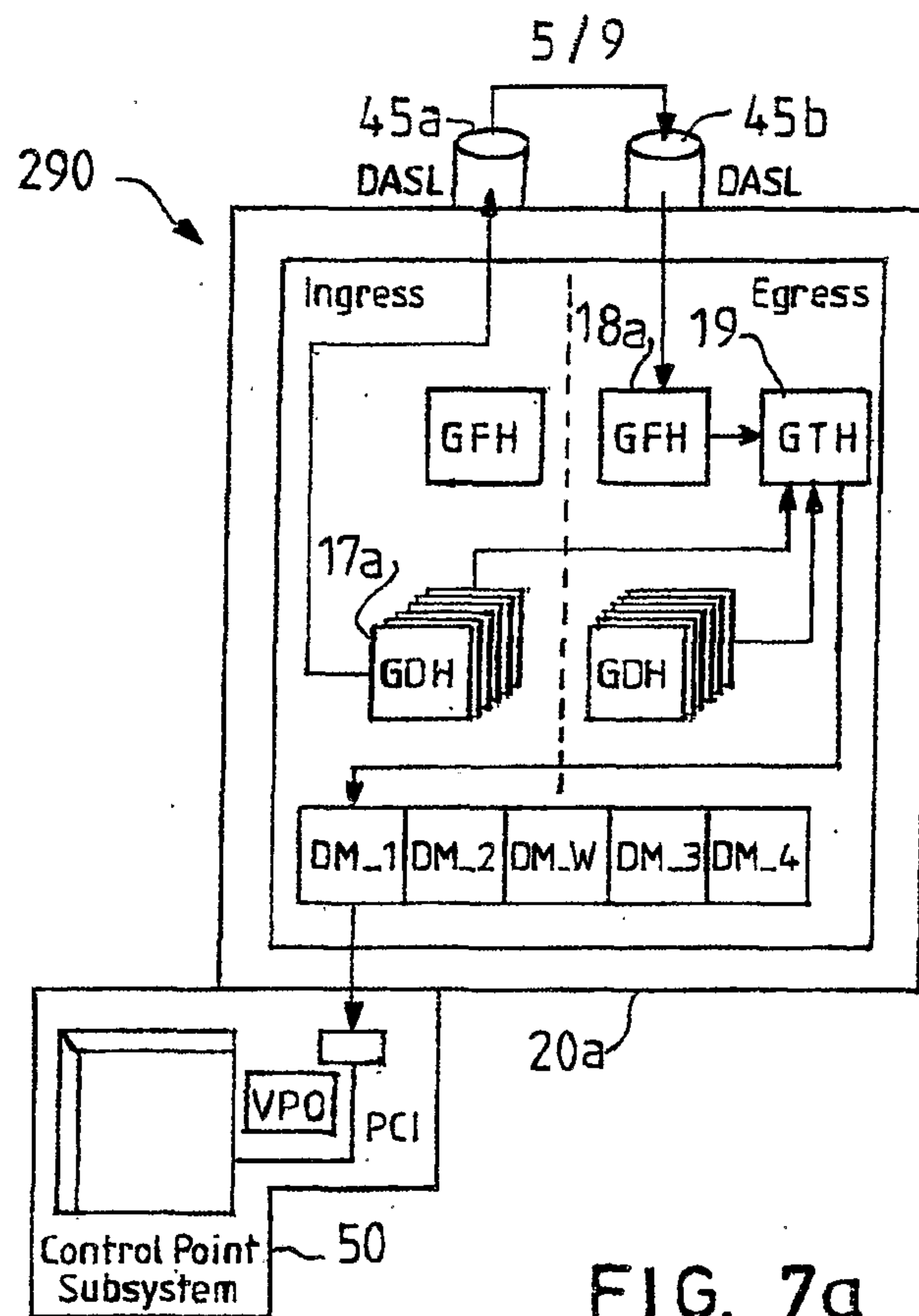


FIG. 5b

FIG. 6aFIG. 6b



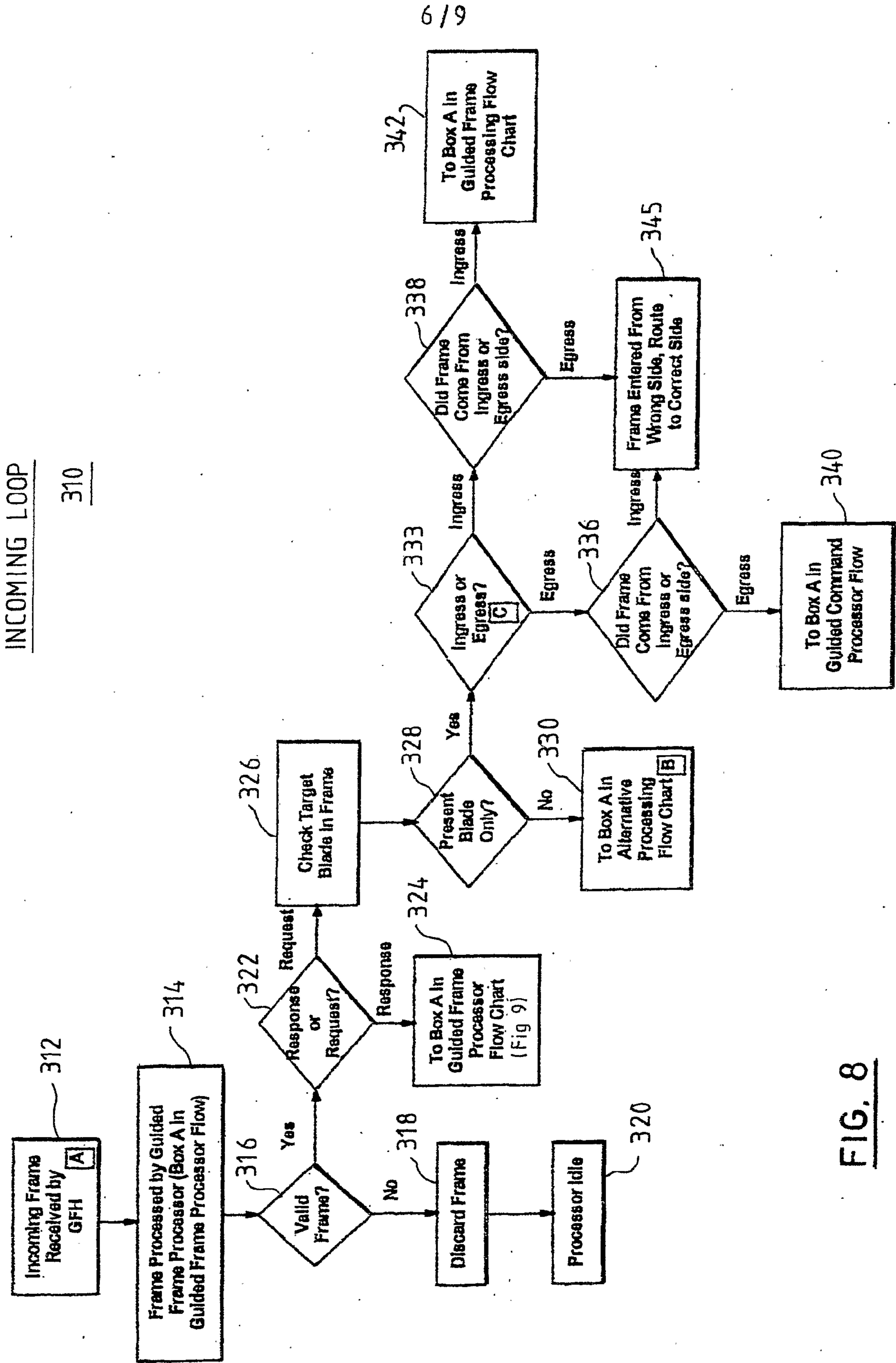
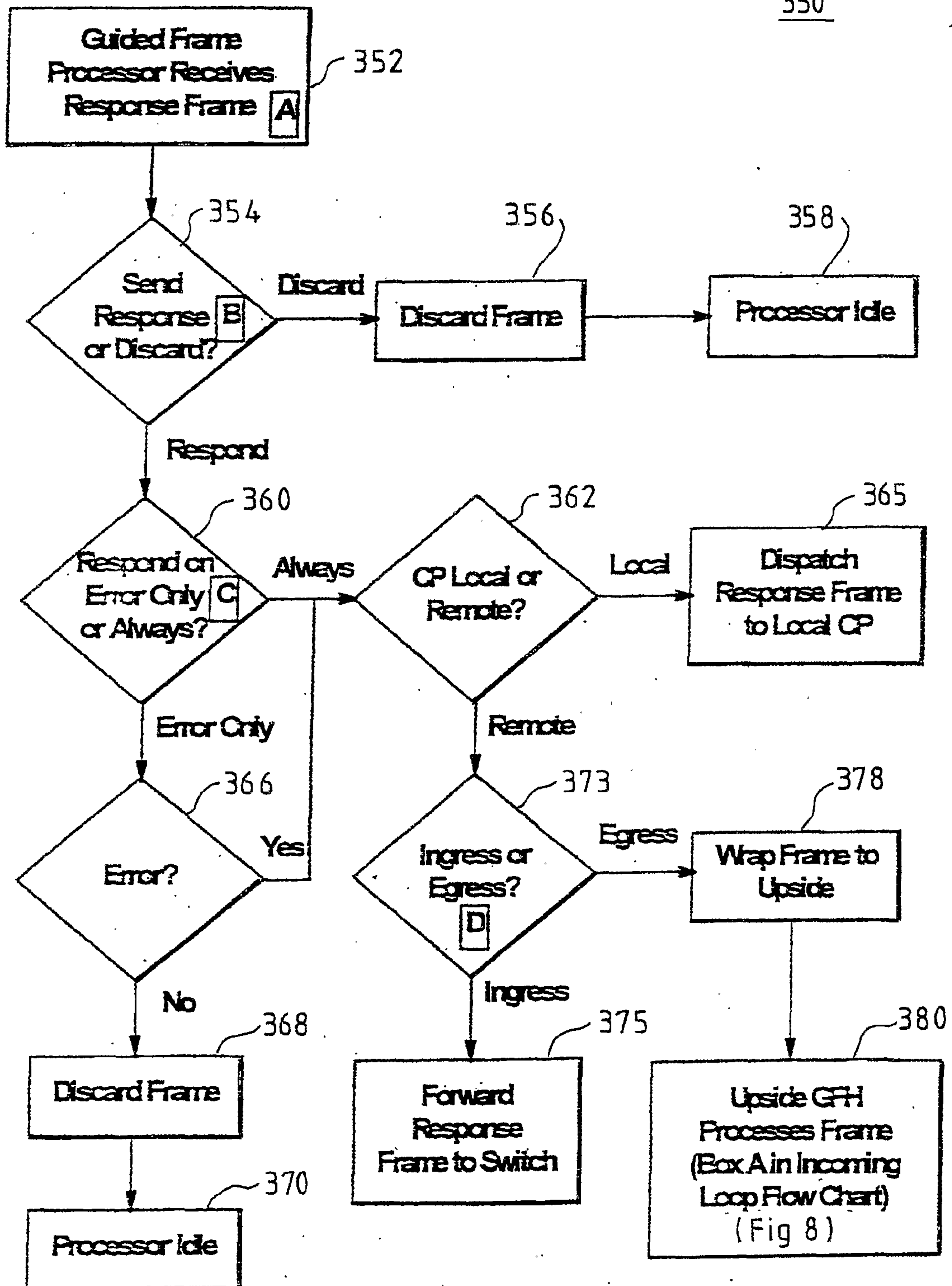
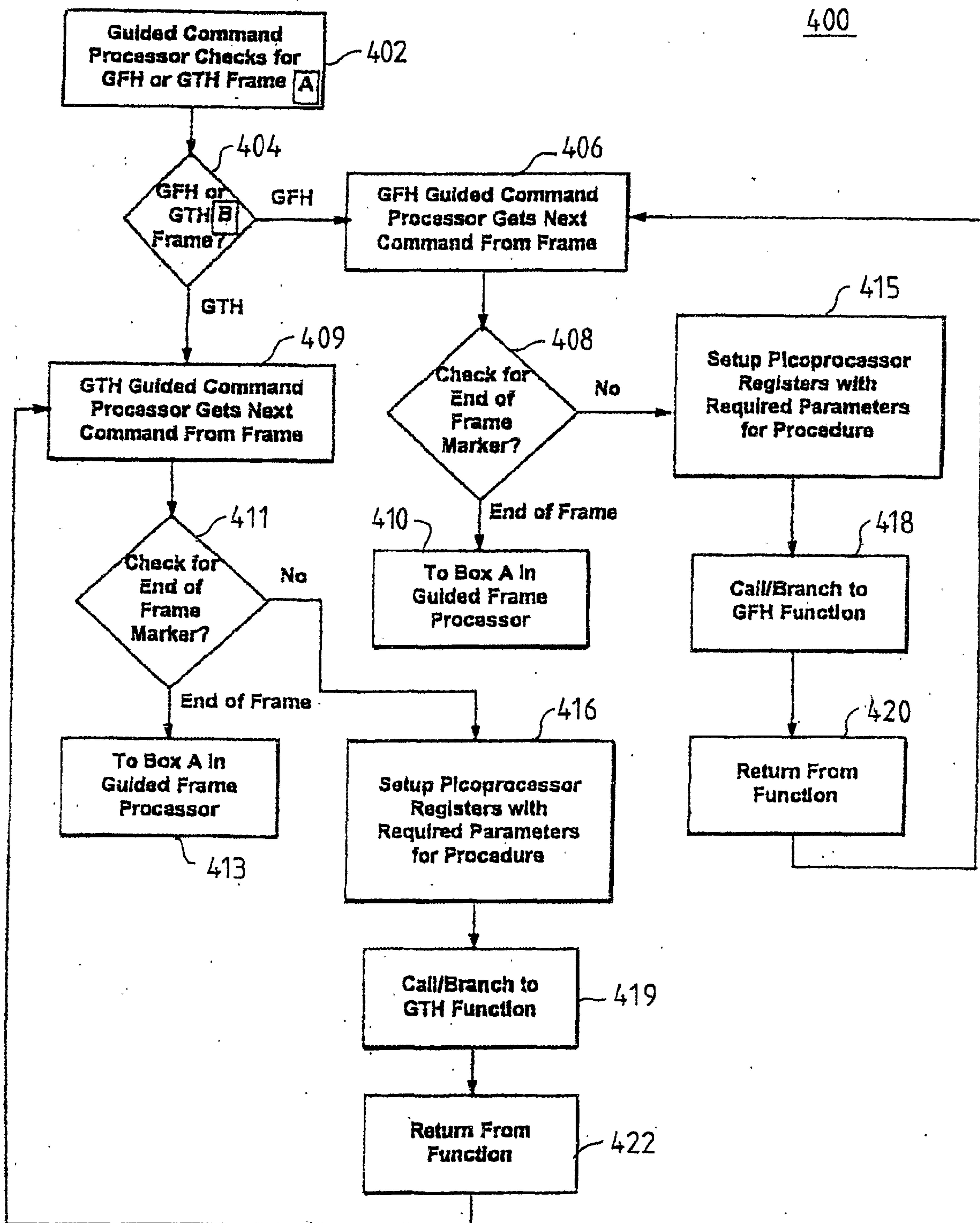


FIG. 8

7/9

GUIDED FRAME PROCESSOR350FIG. 9

8 / 9

GUIDED COMMAND PROCESSING FLOWFIG. 10

ALTERNATIVE PROCESSING FLOW

