



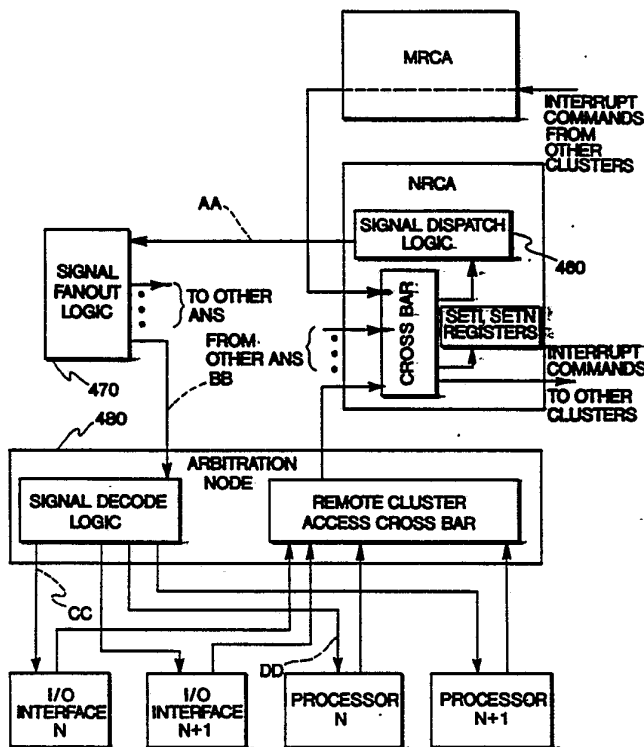
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification <sup>5</sup> : <b>G06F 15/16</b></p>	<p><b>A1</b></p>	<p>(11) International Publication Number: <b>WO 91/20044</b> (43) International Publication Date: 26 December 1991 (26.12.91)</p>
<p>(21) International Application Number: PCT/US91/04060 (22) International Filing Date: 10 June 1991 (10.06.91) (30) Priority data: 536,192 11 June 1990 (11.06.90) US (71) Applicant: SUPERCOMPUTER SYSTEMS LIMITED PARTNERSHIP [US/US]; 1414 West Hamilton Avenue, Eau Claire, WI 54701 (US). (72) Inventors: MILLER, Edward, C. ; 3383 Evergreen Lane, Eau Claire, WI 54701 (US). SPIX, George, A. ; 3309 Westover Lane, Eau Claire, WI 54701 (US). SCHOOLER, Anthony, R. ; S7665 Homestead Road, Eau Claire, WI 54701 (US). BEARD, Douglas, R. ; S10505 Lowes Creek Road, Elewa, WI 54738 (US). PHELPS, Andrew, E. ; 6551 Hillview Road, Eau Claire, WI 54701 (US). SILBEY, Alexander, A. ; 2518 West Princeton Avenue, Eau Claire, WI 54703 (US).</p>		<p>(74) Agents: PEDERSEN, Brad, D.; Dorsey &amp; Whitney, 2200 First Bank Place East, Minneapolis, MN 55402 (US) et al. (81) Designated States: AT (European patent), BE (European patent), CH (European patent), DE (European patent), DK (European patent), ES (European patent), FR (European patent), GB (European patent), GR (European patent), IT (European patent), JP, KR, LU (European patent), NL (European patent), SE (European patent).  <b>Published</b> <i>With international search report.</i></p>

(54) Title: COMMUNICATION EXCHANGE SYSTEM FOR A MULTIPROCESSOR SYSTEM

(57) Abstract

A signaling mechanism for sending and receiving signals to and from any one of all of a plurality of devices, including peripheral controllers (24) and processors (10), in a multiprocessor system. The signaling mechanism includes two switches, a first switch (480) routing a signal command generated by the device to a signal dispatch logic (460) and a second switch (470) for receiving signals generated by the signal dispatch logic and routing the signals to the selected device. The signal dispatch logic (460) receiving the signal command, decodes the destination select value and generates a signal to be sent to the selected device. The signal command includes a destination select value representing a device selectably determined by the device. The signaling mechanism also includes an arbitration mechanism (51) connected to the signal dispatch logic (460) and the first switch for resolving simultaneous conflicting signal commands issued by two or more devices. The signal generated by the signal dispatch logic (460) may include a plurality of bits representing one or more types of predefined signals to be acted upon by the device.



**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	ES	Spain	MG	Madagascar
AU	Australia	FI	Finland	ML	Mali
BB	Barbados	FR	France	MN	Mongolia
BE	Belgium	GA	Gabon	MR	Mauritania
BF	Burkina Faso	GB	United Kingdom	MW	Malawi
BG	Bulgaria	GN	Guinea	NL	Netherlands
BJ	Benin	GR	Greece	NO	Norway
BR	Brazil	HU	Hungary	PL	Poland
CA	Canada	IT	Italy	RO	Romania
CF	Central African Republic	JP	Japan	SD	Sudan
CG	Congo	KP	Democratic People's Republic of Korea	SE	Sweden
CH	Switzerland	KR	Republic of Korea	SN	Senegal
CI	Côte d'Ivoire	LI	Liechtenstein	SU	Soviet Union
CM	Cameroon	LK	Sri Lanka	TD	Chad
CS	Czechoslovakia	LU	Luxembourg	TG	Togo
DE	Germany	MC	Monaco	US	United States of America

5

10

COMMUNICATION EXCHANGE SYSTEM FOR A MULTIPROCESSOR SYSTEMTECHNICAL FIELD

15 This invention relates generally to the field of signaling and interrupt mechanisms for computer and electronic logic systems. More particularly, the present invention relates to a method and apparatus for a signaling mechanism for a multiprocessor system that allows any processor or external interface port to signal any other processor or  
20 external interface port in the multiprocessor system and can resolve simultaneous conflicting signals.

BACKGROUND ART

25 The previously filed parent application entitled CLUSTER ARCHITECTURE FOR A HIGHLY PARALLEL SCALAR/VECTOR MULTIPROCESSOR SYSTEM, PCT Serial No. PCT/US90/07655, describes a new cluster architecture for high-speed computer processing systems, referred to as supercomputers. For most supercomputer applications, the objective is to provide a computer processing system with the fastest  
30 processing speed and the greatest processing flexibility, i.e., the ability to process a large variety of traditional application programs. In an effort to increase the processing speed and flexibility of supercomputers, the cluster architecture for highly parallel multiprocessors described in the previously identified parent application provides an architecture for supercomputers  
35 wherein a multiple number of processors and external interface means can make multiple and simultaneous requests to a common set of shared hardware resources, such as main memory, secondary memory, global

registers, interrupt mechanisms, or other shared resources present in the system.

One of the important considerations in designing such shared-resource, multiprocessor systems is to provide an efficient mechanism for  
5 processors and external interface ports to signal other processors and external interface ports. As used within the present invention, the term signal refers to the operation by which one device (processor or external interface port) indicates to another device that an event has occurred that requires action or intervention by the device being signaled. From a  
10 traditional software perspective, signals are more commonly referred to as interrupts in the sense that the operational flow of the device is interrupted to process the signal.

Many parallel processor architectures implement signals as messages passed through the system on a common bus or channel, such as  
15 in the Intel iPSC Concurrent computer or in the Sequent Balance Series. In this type of architecture, message transmission can take milliseconds for any processor to interrupt another in the system, largely due to the overhead associated with assembling, transmitting, and interpreting a complex message structure. This overhead is a limitation of this type of  
20 signaling architecture.

Other parallel processor architectures do not permit signals to be sent and received by peripheral controllers. In this architecture, processors are dedicated to communicating with input/output devices such that an input/output device can communicate only with the processor to which it  
25 is connected. This restriction limits the flexibility for assigning processors to input/output control tasks.

Another problem with many of the present interrupt mechanisms for multiprocessor systems is that all of the processors in the multiprocessor system are unconditionally interrupted at the completion  
30 of an input/output activity, not just the processors associated with controlling that activity. The disadvantage to this technique is that all programs executing on the multiprocessor system are interrupted which wastes processor resources while the interrupt are being serviced by one of the processors

35 Although the prior art interrupt mechanisms for multiprocessor systems are acceptable under certain conditions, it would be desirable to provide a more effective interrupt mechanism for a multiprocessor

system that was able to allow a process to select any individual interruptable resource to be the targeted handler for servicing a signal. In addition, it would be desirable to provide an interrupt mechanism for the cluster architecture for the multiprocessor system described in the parent application that aids in providing a fully distributed, multithreaded input/output environment.

### SUMMARY OF THE INVENTION

The present invention is a signaling mechanism for a multiprocessor system that allows any processor or peripheral device to signal any other processor or peripheral device in the multiprocessor system and can resolve simultaneous conflicting signals. Unlike present interrupt mechanisms, the signaling mechanism of the present invention provides for targeted signals that include an address that is related to the signal which indicates to the hardware for the signaling mechanism where to direct the particular signal. Simultaneous conflicting signals (i.e., signals targeted to the same peripheral device or processor) are resolved by queuing the signals on a first-come, first-serve basis with an arbitration network determining the priority of simultaneous conflicting signals received during the same clock cycle. The simultaneous conflicting signals are then processed serially based upon the assigned priority.

The present invention requires a very simple code to select a destination to receive a signal and provides a dedicated hardware network for signal distribution that rapidly transmits signals throughout the system. The present invention permits any processor in the system to signal any input/output device, as well as the reverse.

Although it is theoretically possible for all of the devices in a multiprocessor system provided with the present invention to simultaneously issue conflicting signals, the present invention takes advantage of the statistical improbability of this occurrence to optimize the amount of hardware required to process the conflicting signals as compared to the decrease in the overall performance of the multiprocessor system as a result of serially processing such conflicting signals.

The signaling mechanism is accessible throughout the multiprocessor system. All processors and peripheral devices (i.e., secondary memory transfer controllers and peripheral controllers) are able to send and receive signals. In addition, all signals carry two bits of

information that are used by the receiving device to determine what action, if any, should be taken as a result of receiving the signal. These features permit the implementation of a variety of signaling techniques throughout the system. For example, a secondary memory transfer  
5 controller uses type 0 signals as a start command, and uses type 1 signals as a halt command. Because any processor or peripheral device in the system can send a signal to any secondary memory transfer controller, any device can start or stop any secondary memory transfer controller in the system.

An objective of the present invention is to provide a method and  
10 apparatus for a signaling mechanism for a multiprocessor system that allows any processor or external interface port to signal any other processor or external interface port in the multiprocessor system.

Another objective of the present invention is to provide a signaling  
15 mechanism that can resolve simultaneous conflicting signals issued by a plurality of processors or external interface ports.

Another objective of the present invention is to provide a signaling  
mechanism that conveys a plurality of types of signal to the receiving device.

These and other objectives of the present invention will become  
20 apparent with reference to the drawings, the detailed description of the preferred embodiment and the appended claims.

### DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of a single multiprocessor cluster of the  
25 preferred embodiment of the present invention.

Fig. 2 is block diagram of a four cluster implementation of the preferred embodiment of the present invention.

Fig. 3 is a block diagram of showing the implementation of the Fast  
Interrupt mechanism as part of the NRCA means of the preferred  
30 embodiment of the multiprocessor system.

Fig. 4 is an overall block diagram of the input/output architecture of the present invention.

Fig. 5 is a schematic representation showing the signal device  
selection implementation.

35 Fig. 6a is a diagram of the System Mode register.

Fig. 6b is a diagram of the Pending Interrupts register.

Fig. 7 is an overall block diagram of showing the operation of signals in accordance with the present invention.

Fig. 8 shows the condition of the various signals indicated in Fig. 7 at each point in the operational flow of the signaling mechanism.

5 Fig. 9 is a conceptual model of the Secondary Memory Transfer Controller operation.

Fig. 10 shows the SMTC command block.

Figs. 11a and 11b show the command definitions for the SMTC0 and the SMTC1.

10

### DESCRIPTION OF THE PREFERRED EMBODIMENTS

Referring to Fig. 1, the architecture of a single multiprocessor cluster of the preferred embodiment of the multiprocessor system for use with the present invention will be described. The preferred cluster architecture for a highly parallel scalar/vector multiprocessor system is capable of supporting a plurality of high-speed processors 10 sharing a large set of shared resources 12 (e.g., main memory 14, global registers 16, and interrupt mechanisms 18). The processors 10 are capable of both vector and scalar parallel processing and are connected to the shared resources 12 through an arbitration node means 20. Also connected through the arbitration node means 20 are a plurality of external interface ports 22 and input/output concentrators (IOC) 24 which are further connected to a variety of external data sources 26. The external data sources 26 may include a secondary memory system (SMS) 28 linked to the input/output concentrator 24 via a high speed channel 30. The external data sources 26 may also include a variety of other peripheral devices and interfaces 32 linked to the input/output concentrator 24 via one or more standard channels 34. The peripheral devices and interfaces 32 may include disk storage systems, tape storage system, printers, external processors, and communication networks. Together, the processors 10, shared resources 12, arbitration node 20 and external interface ports 22 comprise a single multiprocessor cluster 40 for a highly parallel multiprocessor system in accordance with the preferred embodiment of the present invention.

35 The preferred embodiment of the multiprocessor clusters 40 overcomes the direct-connection interface problems of present shared-memory supercomputers by physically organizing the processors 10, shared resources 12, arbitration node 20 and external interface ports 22

into one or more clusters 40. In the preferred embodiment shown in Fig. 2a and 2b, there are four clusters: 40a, 40b, 40c and 40d. Each of the clusters 40a, 40b, 40c and 40d physically has its own set of processors 10a, 10b, 10c and 10d, shared resources 12a, 12b, 12c and 12d, and external interface ports 22a, 22b, 22c and 22d that are associated with that cluster. The clusters 40a, 40b, 40c and 40d are interconnected through a remote cluster adapter 42 that is a logical part of each arbitration nodes means 20a, 20b, 20c and 20d. Although the clusters 40a, 40b, 40c and 40d are physically separated, the logical organization of the clusters and the physical interconnection through the remote cluster adapter 42 enables the desired symmetrical access to all of the shared resources 12a, 12b, 12c and 12d across all of the clusters 40a, 40b, 40c and 40d.

Referring now to Fig. 3, the physical organization of the Signaling Mechanism in the four-cluster preferred embodiment of the present invention will be described. There are sixteen ports 47 to the global registers 16, signal logic 31, and fast interrupt logic 33 from the thirty-two processors 10 and thirty-two external interface ports 22 in a cluster 40. Each port 47 is shared by two processors 10 and two external interface ports 22 and is accessed over the path 52. A similar port 49 services inter-cluster requests for the global registers 16, fast interrupt logic 31, and signal logic 33 in this cluster as received by the MRCA means 48 and accessed over the path 56. As each request is received at the NRCA means 46, a cross bar and arbitration means 51 direct requests to the appropriate destination. If simultaneous requests come in for access to the SETN registers in the fast interrupt logic 33, for example, these requests are arbitrated for in a pipelined manner by the cross bar and arbitration means 51. The cross bar and arbitration means 51 utilizes a Multiple Request Toggling scheme algorithm. It receives input from sixteen arbitration nodes 44 and one MRCA means 48. An arbitration decision requires address information to select the target register and control information to determine the operation to be performed. This information is transmitted to the NRCA means 46 along with the data. The address and control can be for data to be sent to global registers 16 or to signal logic 31 or the fast interrupt logic 33.

The Multiple Requestor Toggling (MRT) priority system of the preferred embodiment of the present invention allows fair and efficient arbitration of simultaneous multiple requests to common shared



resources by using a simple boolean algorithm to control a variety of switching mechanisms. All requestors are arbitrated in a distributed and democratic fashion by assigning priority to multiple requests on a first-come, first-serve basis with the priority of multiple simultaneous requests being resolved on the basis of a toggling system. The MRT priority scheme is applicable to any system where multiple requestors communicate with a commonly shared resource requiring an arbitration network to resolve simultaneous conflicting requests to that resource. In this case, resolution of conflicts refers to the determination of the order in which requests for access to the common resource are serviced. The MRT priority system is also useful for determining access to multiple shared resources. In this case, part of the MRT priority system, an inhibit matrix, is associated with each one of the multiple shared resources and the plurality of these inhibit matrixes are connected to each of the requestors. Each of the inhibit matrixes per shared resource are connected to a common part, the relative priority state storage means, which maintains the priority of each requestor relative to the others for all shared resources. The relative priority state storage means which stores the relative priority state of every requestor relative to every other requestor. Each cell or bit in the relative priority state storage means represents the relative priority of two requestors. This cell indicates which of the requestors will be granted access in the case of simultaneous resource requests. Each of the cells of the relative priority state storage means are connected to the inhibit matrix. Each cell in the relative priority state storage means drives two gates in the inhibit matrix for that destination. One gate represents requestor x inhibiting requestor y if x is higher priority, while the other gate represents requestor y inhibiting requestor x if y is highest. In this manner, the MRT priority system can be used to control a wide range of switching applications.

Referring now to Fig. 4, an overview of the architecture for the input/output system of the preferred embodiment of present invention will be described. The input/output peripheral devices 32 are connected through the standard channels 34, the input/output concentrator 24 and the external interface ports 22 to the main memory (MM) 14 and global registers 16 and can directly read and write to these shared resources 12 within the same cluster 40, as well as in other clusters 40. The peripheral devices 32 can also read and write to secondary memory (SM) in the

secondary memory system (SMS) 28 associated with the same cluster 40a, for example, but cannot access the SMS 28 in other clusters 40b-40d. It should be noted that a path is not available to allow processors 10 and peripheral devices 32 to directly exchange data. Any such exchanges must  
5 take place through main memory 14, SMS 28 or the global registers 16.

The input/output concentrator (IOC) 24 contains the data paths, switches, and control functions to support data transfers among the various input/output components. In the preferred embodiment, either  
10 eight or sixteen IOC's 24 are physically located within a single input/output chassis 100. Each IOC 24 supports up to eight channel adapters 102 that interface to the standard channels 34 and the peripheral controllers 103, a secondary memory transfer controller (SMTC) 104 that controls a secondary memory port 106 to the high speed channel 30 and the SMS 28, a cluster port 108 that connects to the external interface ports  
15 22, concentrator signal logic 110 that distributes interrupt signals to the channel adapters 102 and the SMTC 104, and a data path crossbar switch 112. Each IOC 24 can read or write a single, 64-bit word in main memory 14 every other clock cycle. The IOC 24 can also read or write a word to the SMS 28 while simultaneously accessing main memory 14.

Each channel adapter 102 contains the functions necessary to  
20 exchange data with a peripheral device controller 103 from an input/output peripheral device 32 over a standard input/output channel 34. The channel adapters 102 access main memory 14, SMS 28 and global registers 16, and send signals to the processors 10 through the IOC 24. The  
25 cross bar switch 112 in the IOC 24 multiplexes access requests among the channel adapters 102 attached to it, routing data to the destination selected by a given transfer. All eight channel adapters 102 requesting data at the maximum rate require the maximum available rate from main memory 14 or the maximum available rate from SMS 28.

The peripheral controllers 103 through the standard channel 34 can  
30 initiate signals by writing the destination select value to the signal interrupt logic 31. A command code is supported by the standard channel 34 that allows a peripheral controller 103 to perform this operation. The SMTC 104 may also transmit signals to peripheral device controllers 103.  
35 Logic in the input/output system initiates the appropriate channel activity when it detects that a signal has been sent to the device associated with any

given channel. This method is used to initiate signals and the action taken in response to a signal varies according to device type.

A destination for the signals is selected by transmitting a destination select value along with the signal. Fig. 5 shows the logical-to-physical mapping for the destination select values. Both processors 10 and IOCs 24 can send and receive signals, in the same and in different clusters 40. The following describes how the contents of the Signal Value are interpreted in the system:

Cluster Select determines which cluster 40 the Signal will be sent to. Logic in the NRCA means 46 and MRCA means 48 determines which cluster 40 is signalled for any value.

Substrate Select determines the physical processor 10 or input/output concentrator 24 which will receive the signal.

Class Select determines which type of device will receive the interrupt. The two bit code is as follows: 0 - processor, 1 - input/output concentrator, 2- secondary memory transfer controller, and 3 - reserved.

Channel Select. When an input/output concentrator 24 is specified in the Class Select field, bits 4 through 2 address a channel adapter on the IOC 24 selected in the Substrate Select field. When the secondary memory transfer controller is specified in the Class Select field, bit 2 selects which secondary memory transfer controller in an input/output concentrator means 26 will be interrupted: 0 - The Main Memory to Secondary Memory Transfer Controller is signalled, 1 - the Secondary Memory to Main Memory Transfer Controller will be signalled. This field is ignored for all other class selections.

Type Select determines which type of signal is to be transmitted. The signal type is captured at the destination device. The effect of different types of signals is device dependent.

Processors 10 generate Signals through the Signal instruction. For signals generated by the Signal instruction, the value in the S register selected by the Signal instruction is interpreted as the destination select value. Signals are received by the processors 10 as interrupt requests. Referring to Figs. 6a and 6b, the signal are masked by the Disable Type bits (DTO-3) in the System Mode register. Masks for the Interval Timer and Fast Interrupt requests are also located in the System Mode register. Pending interrupts are captured in the Pending Interrupt (PI) control register. A bit in the PI register corresponds to each type of interrupt. An

incoming signal sets the appropriate PI register bit and causes an interrupt if the SM mask for that bit is not set. PI bits are cleared by the interrupt handler code after recognizing the interrupts.

Referring now to Fig. 7, a logical block diagram shows the operation of signals (interrupts) within the present invention. Processors 10 may initiate signals by executing the Signal instruction. The Signal instruction causes the contents of the referenced S-register to be sent to the NRCA means 46 through the arbitration node 44. Similarly, peripheral devices (i.e., peripheral controllers 103 and SMTCs 104) initiate signals by sending a command and signal value to NRCA means 46 through the port 47 in the arbitration node 44. The NRCA means 46 examines the cluster select bits in the signal value and directs the signal to the appropriate cluster.

If the signal is directed to the cluster 40 that the NRCA means 46 is currently located, the NRCA means 46 will direct the signal to the global register crossbar 51 in that NRCA means 46. If the signal is directed to another cluster 40, the NRCA means 46 will send the signal to that cluster 40 over the inter-cluster communication paths 58 via the MRCA means 48. The global register crossbar 51 will direct any signal to the signal dispatch logic 460. Fig. 8 relates to Fig. 7 by showing the signal codes as transmitted on the indicated paths (e.g., AA, BB, etc.) in the signal mechanism shown in Fig. 7

Once the signal value has reached the signal dispatch logic 460 in the NRCA means 46, it is dispatched from there using the signal fanout logic 470. A 13-bit code, shown as AA in Fig. 8, is sent from the dispatch logic 460 to the fanout logic 470. The code is the same as the signal select value, but does not have the cluster select bits attached. They are no longer necessary at this point since the value has already been directed to the proper cluster 40.

The signal fanout logic 470 decodes the substrate select field and sends a 9-bit signal code, shown as BB in Fig. 8, to the arbitration node 44 of the processor 10 or external interface port 22 being signaled. Separate signal buses connect the fanout logic 470 with each arbitration node 44.

Additional signal decode logic 480 within the arbitration node 44 further decodes the 9-bit signal code. A three-bit code, shown as DD in Fig. 8 is presented to each of the processors 10 attached to each arbitration node 44. A seven-bit code, shown as CC in Fig. 8 is presented to each external

interface ports 22 attached to the arbitration node 44 for further transmission to the IOC 24.

The processors 10 further decode the signal value into the four types of signal and sets the appropriate bit in the PI register. If the corresponding interrupt disable bits are cleared in the SM register, processor instruction will be interrupted when the interrupt bit is set in the PI register.

The IOC 24 further decodes the 7-bit signal code sent from the arbitration node 44 into individual signals that are sent to the channels and the SMTCs.

Referring now to Fig. 9, the Secondary Memory Transfer Controller (SMTC) 104 of the IOC 24 of the preferred embodiment is described. In the preferred embodiment, the SMTC 104 controls transfers to the SMS 28. The only addressable unit in the SMS 28 is a block of 32, 64-bit words. Transfers are constrained to begin on a block boundary. Requests for secondary memory transfers (reads or writes) may be initiated by either the channel adapters 102 or the SMTC 104. Transfers to the channel adapters 102 and to the cluster port 108 may proceed simultaneously. Error detection and correction is done at the SMTC 104. In the preferred embodiment, the SMTC 104 consists of two independent controllers 104a and 104b, one for moving data from main memory 14 to the SMS 28 and the other for moving data from the SMS 28 to main memory 14. The controllers accept commands in the form of command blocks that are constructed in main memory 14. The command blocks provide the starting address in main memory 14, the starting address in secondary memory 28, the increment on the base address in secondary memory 28, the number of 32-word blocks to be moved, and the direction of the transfer. Transfer size can range between 1 and (memory size/32) blocks.

As illustrated in Fig. 9, command execution is initiated by sending a signal 400 to the SMTC 104a or 104b. The preferred embodiment has up to 32 pairs of SMTCs 104a and 104b in a fully-configured cluster 40. The particular SMTCs 104 within a cluster 40 are selected by the Signal "substrate select" field in the SMTC Command word that is part of the SMTC Command Block as shown in Fig. 10. Separate signals 400a and 400b initiate transfers in each direction. The SMTC0 at channel select address 0 controls transfers in the direction from main memory 14 to secondary memory 8. The SMTC1 at channel select address 1 controls transfers in the direction from secondary memory 28 to main memory 14.

SMTC selection is based on the LSB of the channel select field so that odd-numbers select SMTC1 and even numbers select SMTC0.

The SMTC 104 recognizes four signal types. The response of the SMTC 104a and 104b to receiving each of the four signal types is described for the preferred embodiment in Table I.

Table I

5	Type 00 -	Fetch command block and start the specified transfer.
10	Type 01 -	Stop the transfer in progress. Transfer status is reported when the SMTC actually halts the transfer. A completion interrupt will be generated if requested in the command packet that started the transfer. If no transfer is in process when a Stop signal is received, no action is taken by the SMTC.
15	Type 10 -	Reserved
20	Type 11 -	Reserved

At the conclusion of a transfer, a status word is written back to the command block in main memory 14 and an optional completion interrupt can be generated. Interrupt generation is specified by the contents of the command block that originated the transfer. The target of the completion interrupt is also determined by the command block.

The SMTC Command block format is shown as it appears in main memory 14. The following are definitions of the command block words and are defined in Table II.

Table II

30	bit 0 - 3	command field
	bit 4	interrupt on operation complete
	bit 5 - 8	transfer priority
	bit 9 - 63	unused

SMTC command field contains bits to indicate that either a transfer operation or a reset operation is to be done. It also determines whether an interrupt is generated on completion. The command field for SMTC0 is

defined in Fig. 11a and the command field for SMTC1 is defined in Fig. 11b. The "interrupt on operation complete" field (command word bit 4) directs the SMTC 104 to issue a Signal 400 at the conclusion of the requested operation. A Signal 400 is sent if this bit is set to one in the command block. No Signal is sent if this bit is zero. The device that will receive the Signal 400, if requested, is determined by the contents of word six of the command block (Signal Device Selection).

SMS FBA is the first address in SMS 28 to begin transferring data to or from. Only bits 31 - 0 are used during the transfer. Bits 63 - 32 are not used by the SMTC 104 and are ignored. The 32-bit value is interpreted as a block address in secondary memory 28. A value of 00000000 will point to the first word of the first block in secondary memory 28. A value of 00000001 will point to the first word in the second block in secondary memory 28.

SMS NBLOCKS is the number of 32-word blocks to transfer. Only bits 31 - 0 are used during the transfer. Bits 63 - 32 are not used by the SMTC and are ignored. A "one" in this field will transfer one 32-word block. It will be noted that a zero in this field will transfer  $2^{32}$  blocks (approximately one Terabyte).

SMS BLKINCR is the block address increment between adjacent data blocks moved by the SMTC 104. Only bits 31 - 0 are used during the transfer. Bits 63 - 32 are not used by the SMTC and are ignored. This concept is shown in Fig. 10. An increment of "one" will transfer a contiguous block of secondary memory 28. It will be noted that if zero is placed in SMS BLKINCR, then the same block will be transferred for NBLOCKS. If  $(\text{SMS FBA} + (\text{NBLOCK} * \text{SMS BLKINCR} * 32))$  is greater than the memory available in SMS 28, the transfer will wrap-around into available memory.

MM FWA is the first word in main memory 14 to begin transferring to or from. Only bits 33 - 0 are used during the transfer, bits 63 - 34 are not used by the SMTC 104 and are ignored. The 34-bit value is interpreted as a word address in main memory 14. If  $(\text{MM FWA} + (\text{NBLOCK} * 32))$  is greater than the memory available in main memory 14, the transfer will wrap-around into available memory.

TRANSFER STATUS. This area of the SMTC command area is used for reporting of transfer completion information and error reporting. Bits are assigned as shown in Table III.

Table III

	Bit 0	operation complete
	Bit 1	double-bit error in data transfer
	Bit 2	invalid command
5	Bit 3	parity error
	Bit 4	command fetch error
	Bit 5	sequence error (another start has been received while a previous transfer is still in progress)

10           Detection of a double-bit or parity error during the command block fetch causes the SMTC to report the error in the command block status word, but no transfer is started. Errors detected during a data transfer halt the transfer in progress.

15           SIGNAL DEVICE SELECTION contains an address of a device to be signalled upon completion of the operation. If a completion signal is specified in the SMTC command field (bit 4), the SMTC 104 uses this value to select the device to be signalled and the type of signal to be issued.

20           Although the description of the preferred embodiment has been presented, it is contemplated that various changes could be made without deviating from the spirit of the present invention. Accordingly, it is intended that the scope of the present invention be dictated by the appended claims rather than by the description of the preferred embodiment.



**CLAIMS**

1. A signaling mechanism for sending and receiving signals to and from any one of all of a plurality of devices, including peripheral  
5 controllers and processors, in a multiprocessor system, the signaling mechanism comprising:  
first switch means operably connected to the devices for routing a signal command generated by the device, the signal command having a destination select value representing a device  
10 selectably determined by the device;  
signal dispatch logic means operably connected to the first switch means for receiving the signal command, decoding the destination select value and generating a signal to be sent to the selected device; and  
15 second switch means operably connected to the signal dispatch logic and to the devices for receiving the signals generated by the signal dispatch logic and routing the signals to the selected device.
2. The signaling mechanism of claim 1 wherein the processors further  
20 include register means for masking the external interrupt signal.
3. The signaling mechanism of claim 1 further comprising arbitration means operably connected to the signal dispatch logic and the first switch means for resolving simultaneous conflicting signal commands issued by two or more devices.
- 25 4. The signaling mechanism of claim 1 wherein the signal generated by the signal dispatch logic further comprises a plurality of bits representing one or more types of signals that are predefined signals to be acted upon by the device.
5. The signaling mechanism of claim 3 further comprising storage  
30 means operably connected to the arbitration means for queuing a plurality of signal commands.
6. The signaling mechanism of claim 3 wherein the arbitration means is comprised of a multiple request toggling system.
7. A highly parallel computer processing system, comprising:  
35 C multiprocessor clusters operably connected to one another, wherein C is an integer between 2 and 256, inclusive, each multiprocessor cluster comprising:

shared resource means for storing and retrieving data and control information,

5 P processor means for performing computer processing of data and control information, wherein P is an integer between 2 and 256, inclusive;

Q external interface means for transferring data and control information between the shared resource means and one or more external data sources, wherein Q is an integer between 2 and 256, inclusive;

10 Z arbitration node means operably connected to the processor means, the external interface means, and the shared resource means for symmetrically interconnecting the processor means and the external interface means with the shared resource means, wherein Z is an integer between 1 and 128, inclusive, and the ratio of P to Z is greater than or equal to 2; and

15 remote cluster adapter means operably connected to remote cluster adapter means in all other of the multiprocessor clusters for allowing the arbitration node means of the multiprocessor cluster to access the shared resource means of all other of the multiprocessor clusters and for allowing all other of the multiprocessor clusters to access the shared resource means of the multiprocessor cluster,

20 the shared resource means including a signaling mechanism for sending and receiving signals to and from any one of all of a set of devices comprising all of the processor means and external interface means which may be directly accessed by the devices of the multiprocessor cluster and which may be accessed by the devices of all other of the multiprocessor clusters through the remote cluster adapter means.

8. The highly parallel computer processing system of claim 7 wherein the signaling mechanism comprises:

35 first switch means operably connected to the arbitration node means for routing a signal command generated by a device, the signal command having a destination select value representing a device selectably determined by the device;

signal dispatch logic means operably connected to the first switch means for receiving the signal command, decoding the destination select value and generating a signal to be sent to the selected device; and

5 second switch means operably connected to the signal dispatch logic and to the devices for receiving the signals generated by the signal dispatch logic and routing the signals to the selected device.

9. A multiprocessor cluster for a highly parallel computer processing system, the multiprocessor cluster adapted for connection to other similar multiprocessor clusters in the highly parallel computer processing system, the multiprocessor cluster comprising:

shared resource means for storing and retrieving data and control information;

15 P processor means for performing computer processing of data and control information, wherein P is an integer between 2 and 256, inclusive;

20 Q external interface means for transferring data and control information between the shared resource means and one or more external data sources, wherein Q is an integer between 2 and 256, inclusive; and

Z arbitration node means operably connected to the processor means, the external interface means, and the shared resource means for symmetrically interfacing the processor means and the external interface means with the shared resource means, wherein Z is an integer between 2 and 128, inclusive, and the ratio of P to Z is greater than or equal to 2,

25 the shared resource means including a signaling mechanism for sending and receiving signals to and from any one of all of a set of devices comprising all of the processor means and external interface means which may be directly accessed by the devices of the multiprocessor cluster and which may be accessed by the devices of all other of the multiprocessor clusters through the remote cluster adapter means.

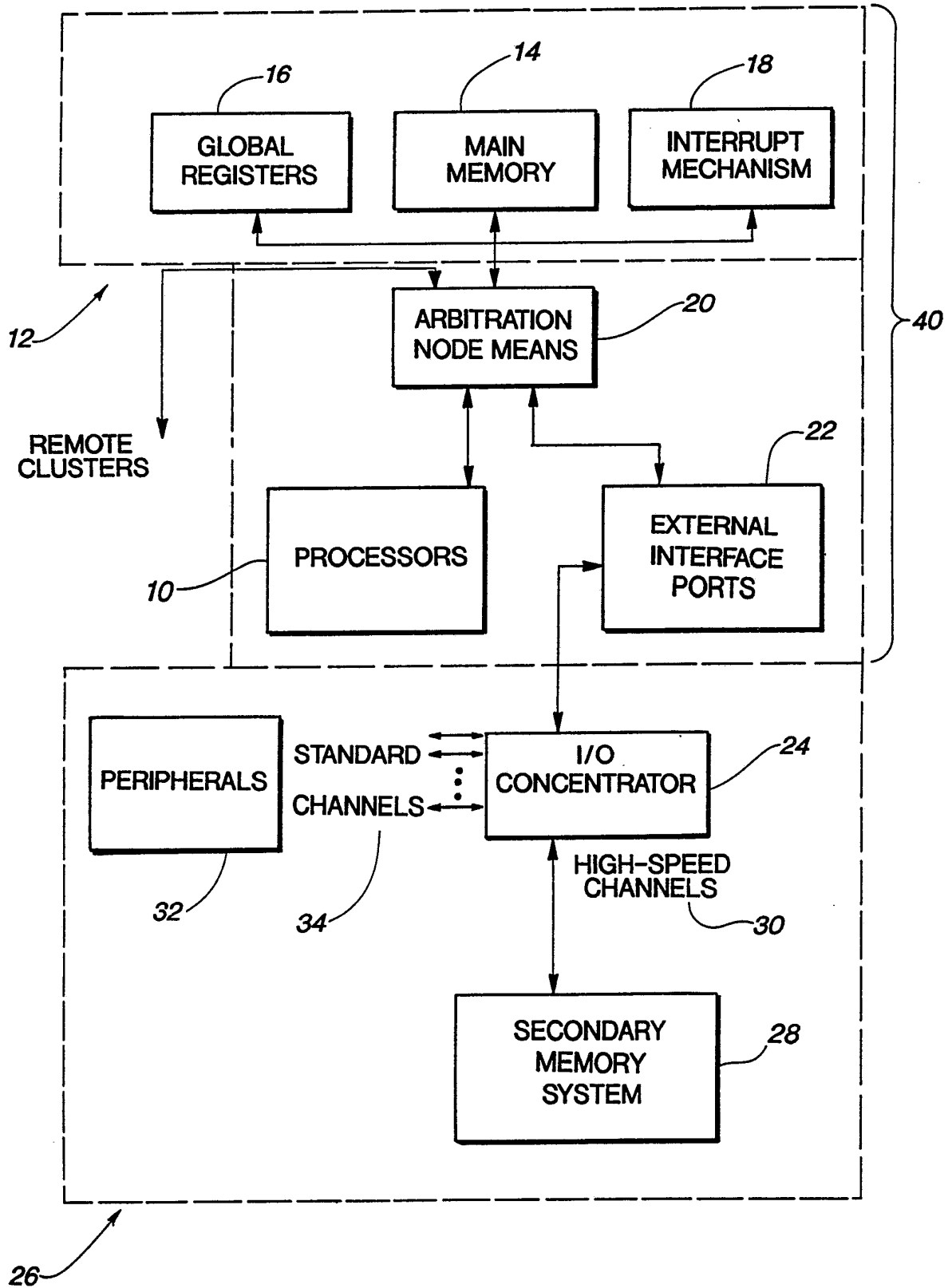
35 10. The highly parallel computer processing system of claim 9 wherein the signaling mechanism comprises:

first switch means operably connected to the arbitration node means for routing a signal command generated by a device, the signal command having a destination select value representing a device selectably determined by the device;

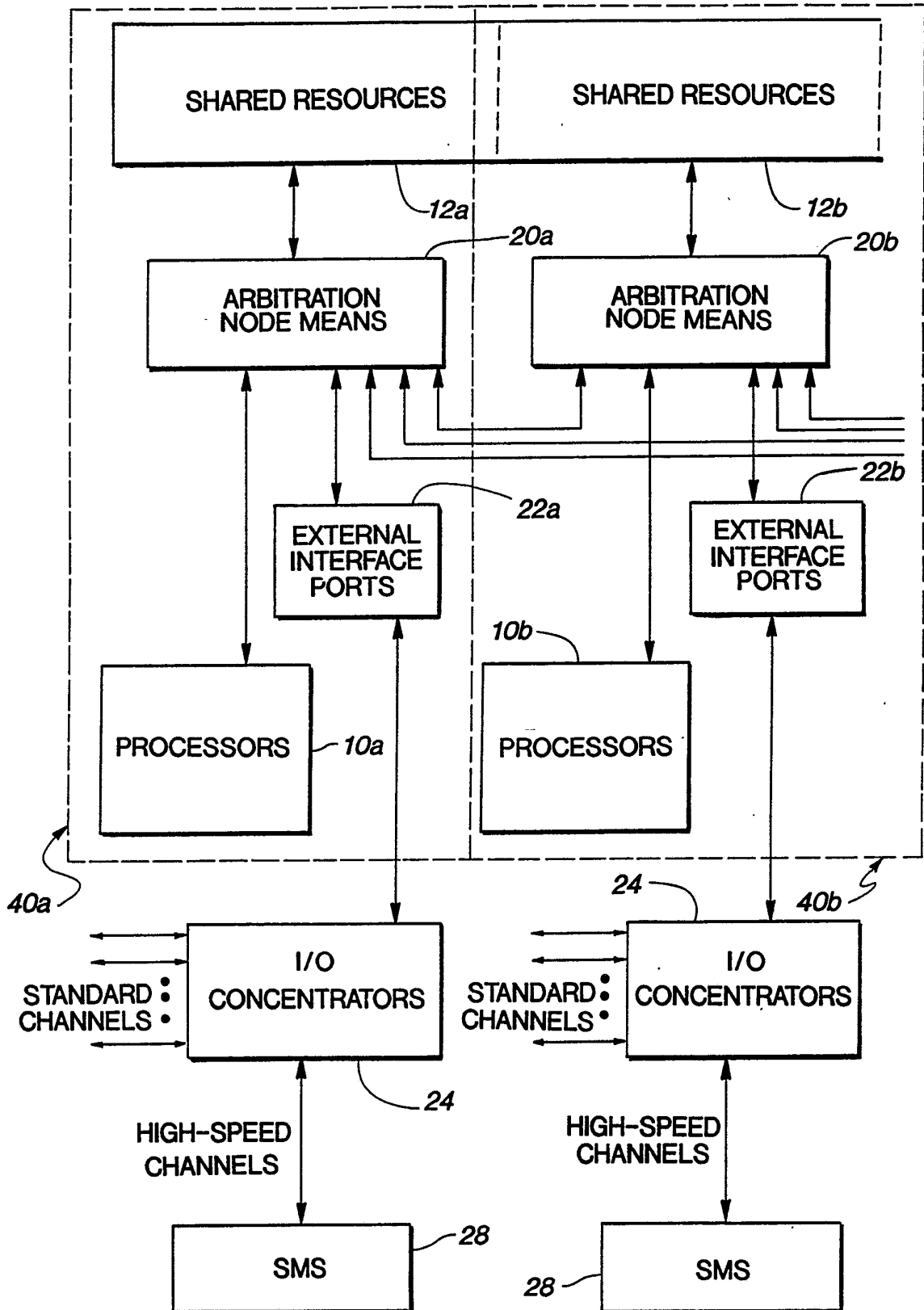
5            signal dispatch logic means operably connected to the first switch means for receiving the signal command, decoding the destination select value and generating a signal to be sent to the selected device; and

10           second switch means operably connected to the signal dispatch logic and to the devices for receiving the signals generated by the signal dispatch logic and routing the signals to the selected device.

Fig. 1



2/13 Fig. 2a



3/13  
*Fig. 2b*

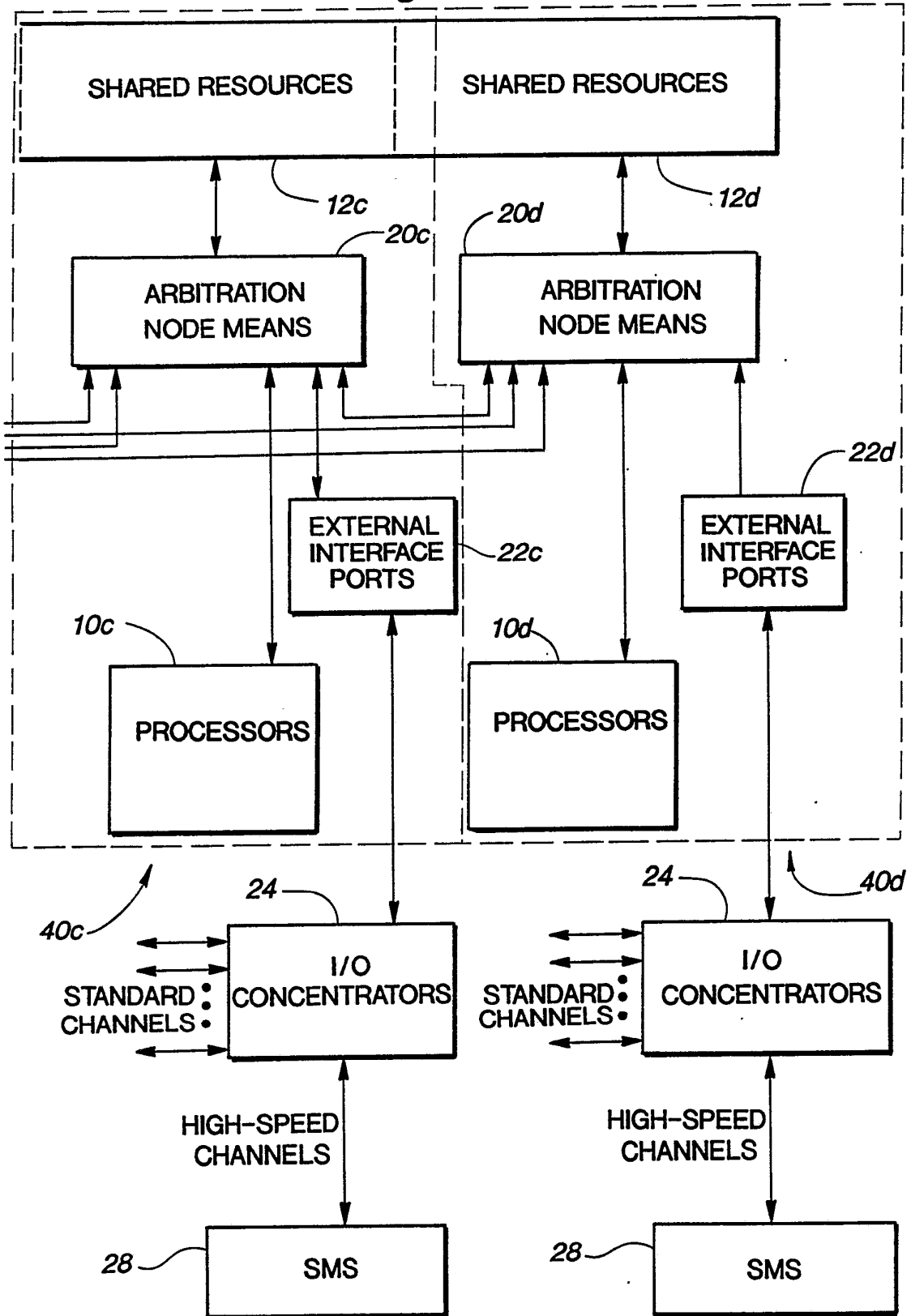


Fig. 3

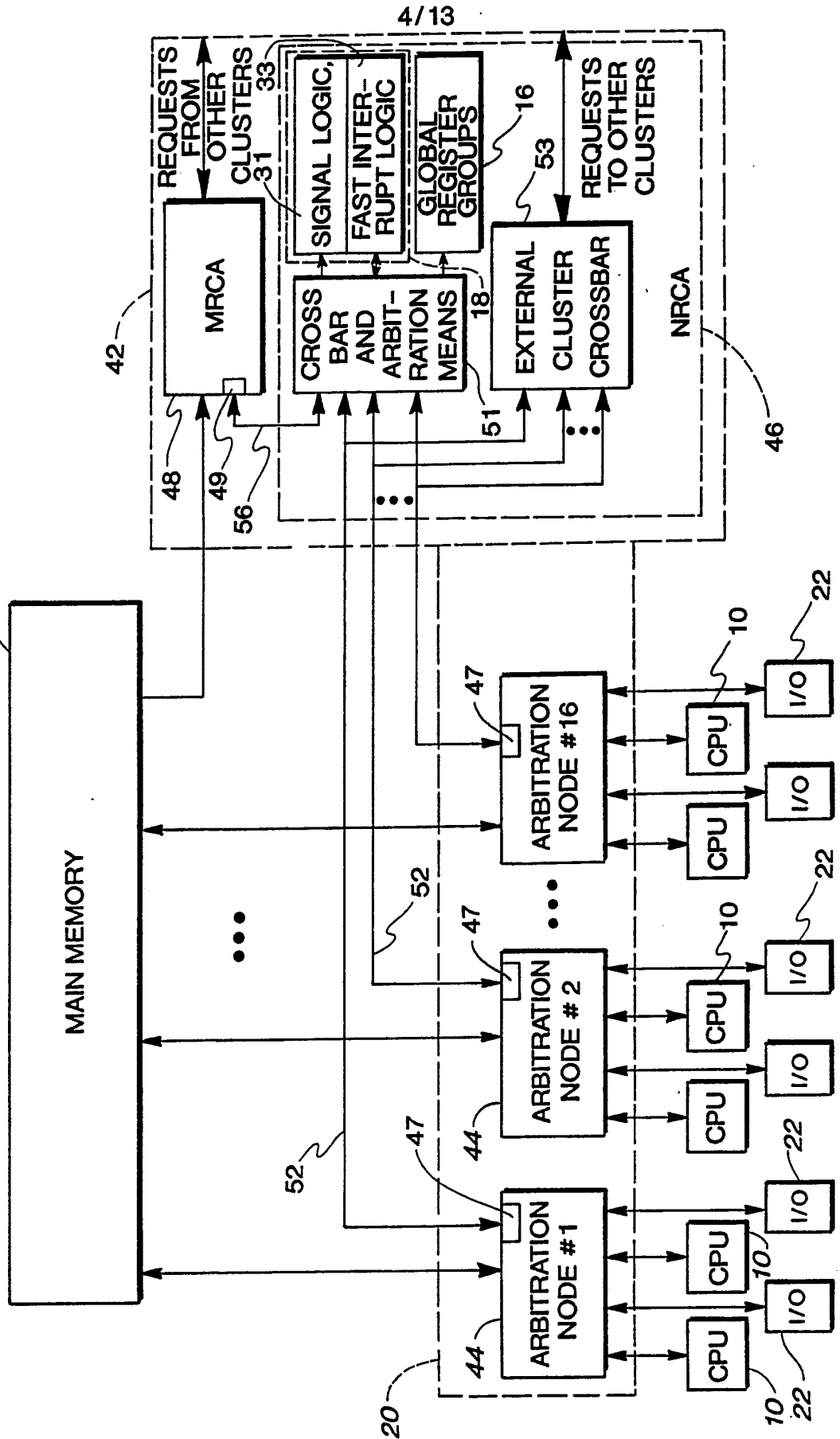




Fig. 4a 5/13

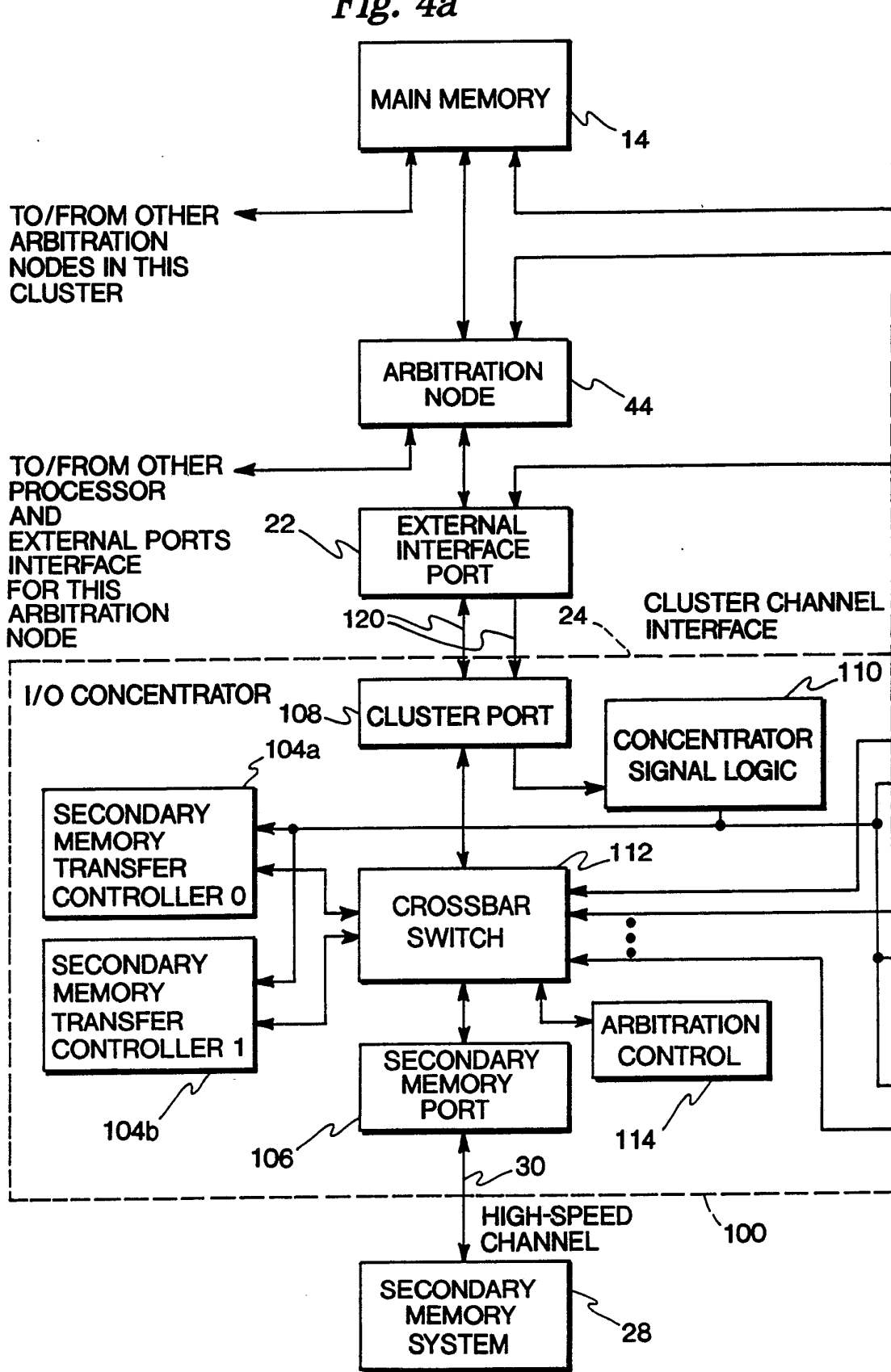
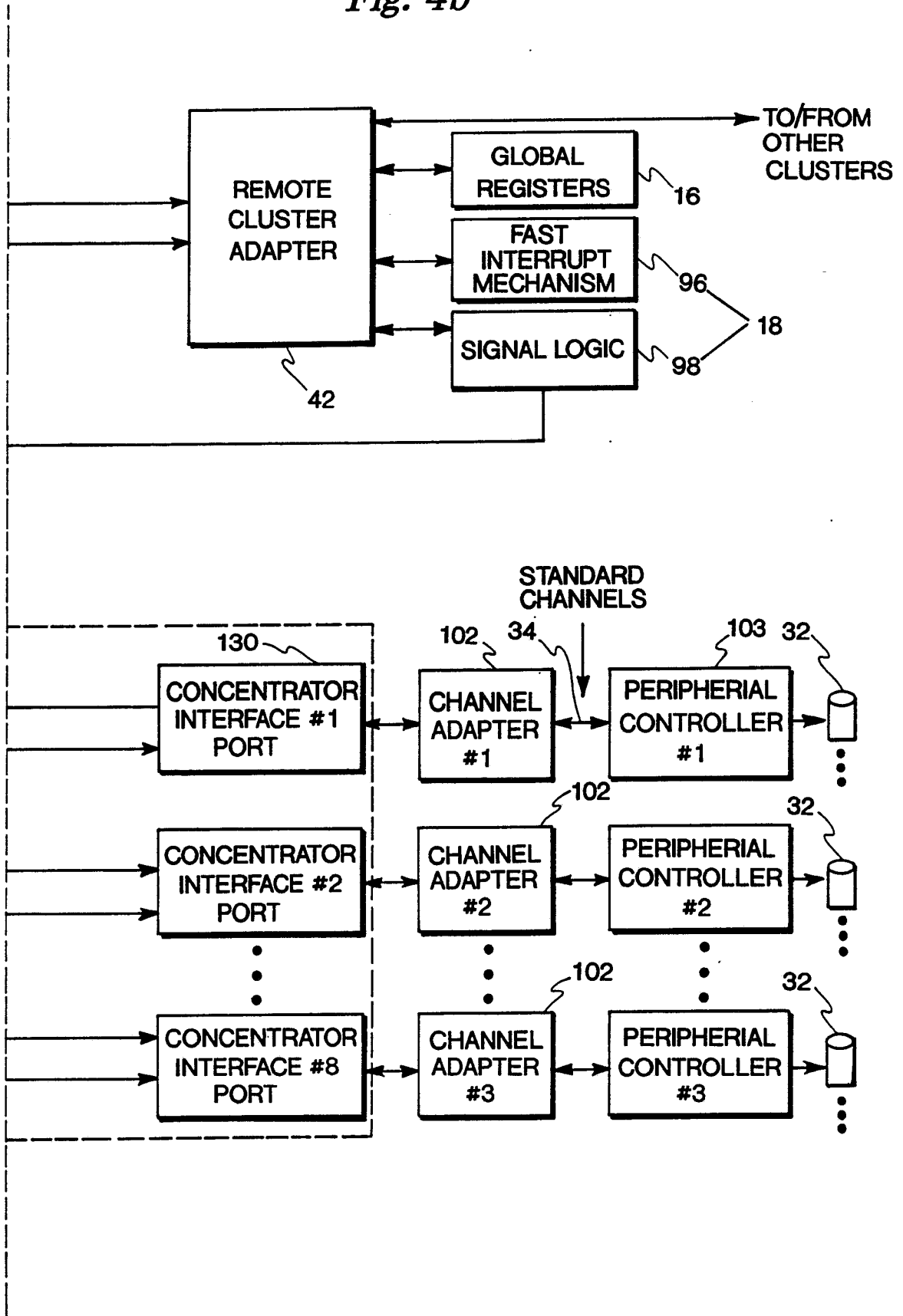
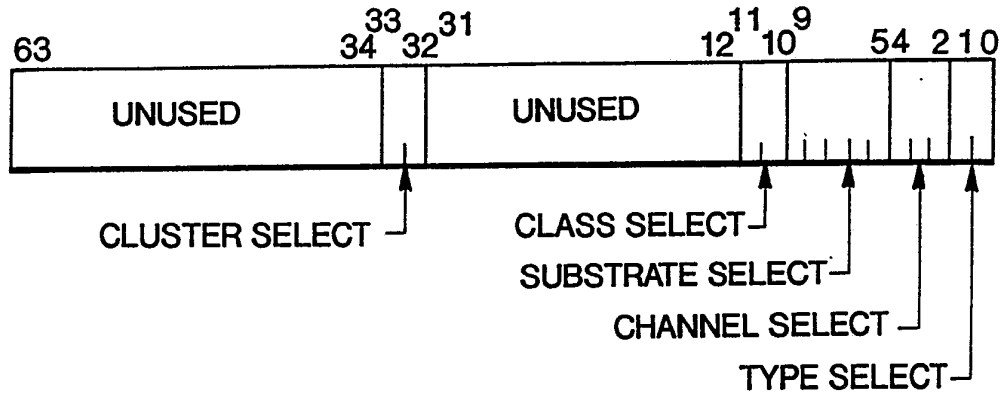


Fig. 4b



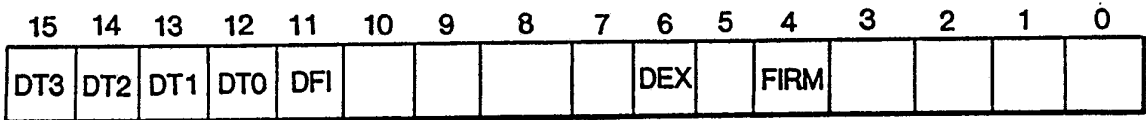
*Fig. 5* <sup>7/13</sup>

SIGNAL DEVICE SELECTION



*Fig. 6a*

SM: SYSTEM MODE (16 BITS)  
NOT USER WRITABLE.



THIS REGISTER IS SAVED IN OSM AND THEN SET TO ALL ONES ON AN INTERRUPT, SYSTEM CALL, OR EXCEPTION. IT IS RESTORED FROM OSM BY THE RTT INSTRUCTION.

**FIRM:** 1=FAST INTERRUPT REQUESTS MASKED, 0=ENABLE. THIS BIT PREVENTS AN EXCEPTION (AS REPORTED IN EXCEPTION STATUS) OR A FAIR INSTRUCTION FROM CAUSING A FAST INTERRUPT REQUEST.

**DEX:** 1=DISABLE CONTEXT SWITCH ON ES REGISTER NON-ZERO, 0=ENABLE.

**DFI:** 1=DISABLE INCOMING FAST INTERRUPT, 0=ENABLE.

**DT0:** 1=DISABLE TYPE 0 SIGNALS. 0=ENABLE.

**DT1:** 1=DISABLE TYPE 1 SIGNALS, 0=ENABLE.

**DT2:** 1=DISABLE TYPE 2 SIGNALS, 0=ENABLE.

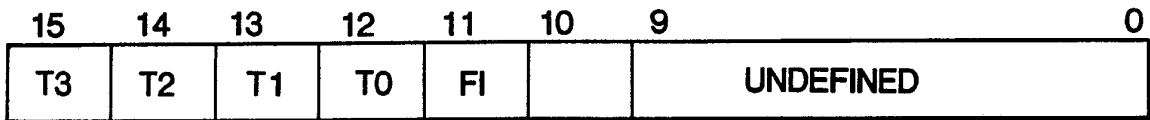
**DT3:** 1=DISABLE TYPE 3 SIGNALS, 0=ENABLE.

8/13

*Fig. 6b*

PI: PENDING INTERRUPTS (16 BITS)  
NOT USER WRITABLE.

THIS REGISTER SHOWS THE STATUS OF INCOMING INTERRUPTS.



- FI:**            1=FAST INTERRUPT SIGNAL HAS BEEN RECEIVED,  
                    0=NO SIGNAL
  
- T0:**            1=TYPE 0 SIGNAL HAS BEEN RECEIVED, 0=NO SIGNAL
- T1:**            1=TYPE 1 SIGNAL HAS BEEN RECEIVED, 0=NO SIGNAL
- T2:**            1=TYPE 2 SIGNAL HAS BEEN RECEIVED, 0=NO SIGNAL
- T3:**            1=TYPE 3 SIGNAL HAS BEEN RECEIVED, 0=NO SIGNAL

WHEN READ, THE REGISTER SHOWS WHICH INTERRUPTS ARE PENDING. READING THE REGISTER CLEARS ALL BITS TO ZERO. WHEN WRITTEN, WRITING A ZERO HAS NO EFFECT ON THE STATE OF ANY BIT. WRITING A ONE IN ANY BIT POSITION WILL SET THAT BIT TO ONE.

*Fig. 4*

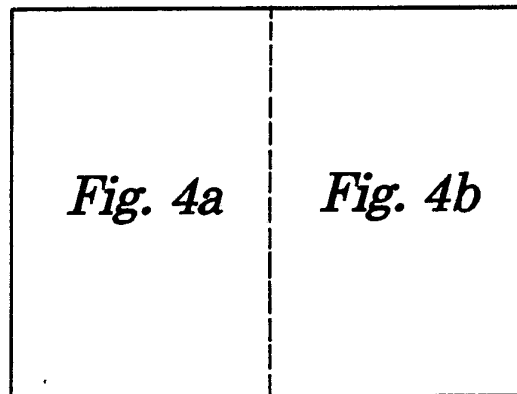


Fig. 7

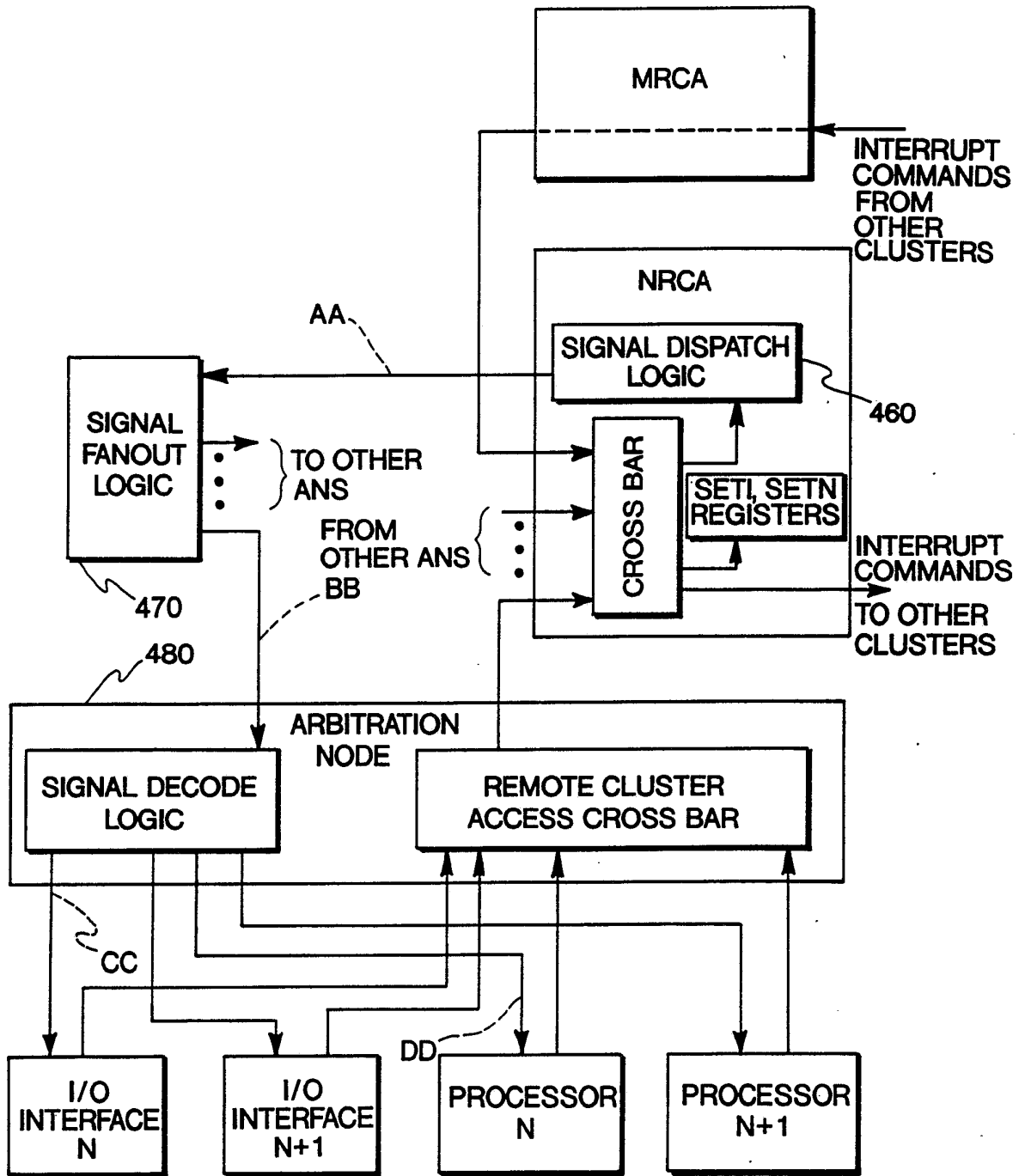
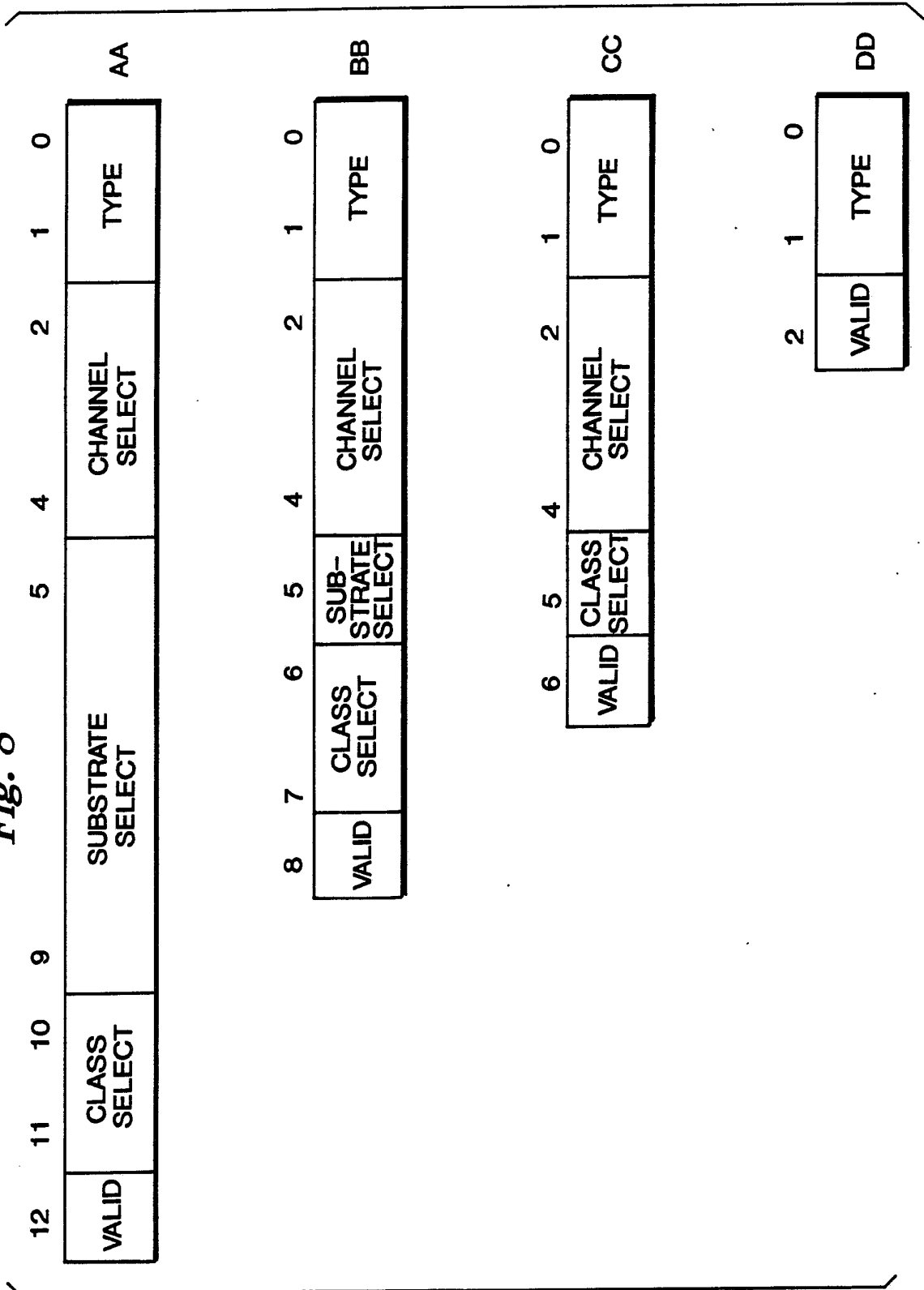


Fig. 8



11/13

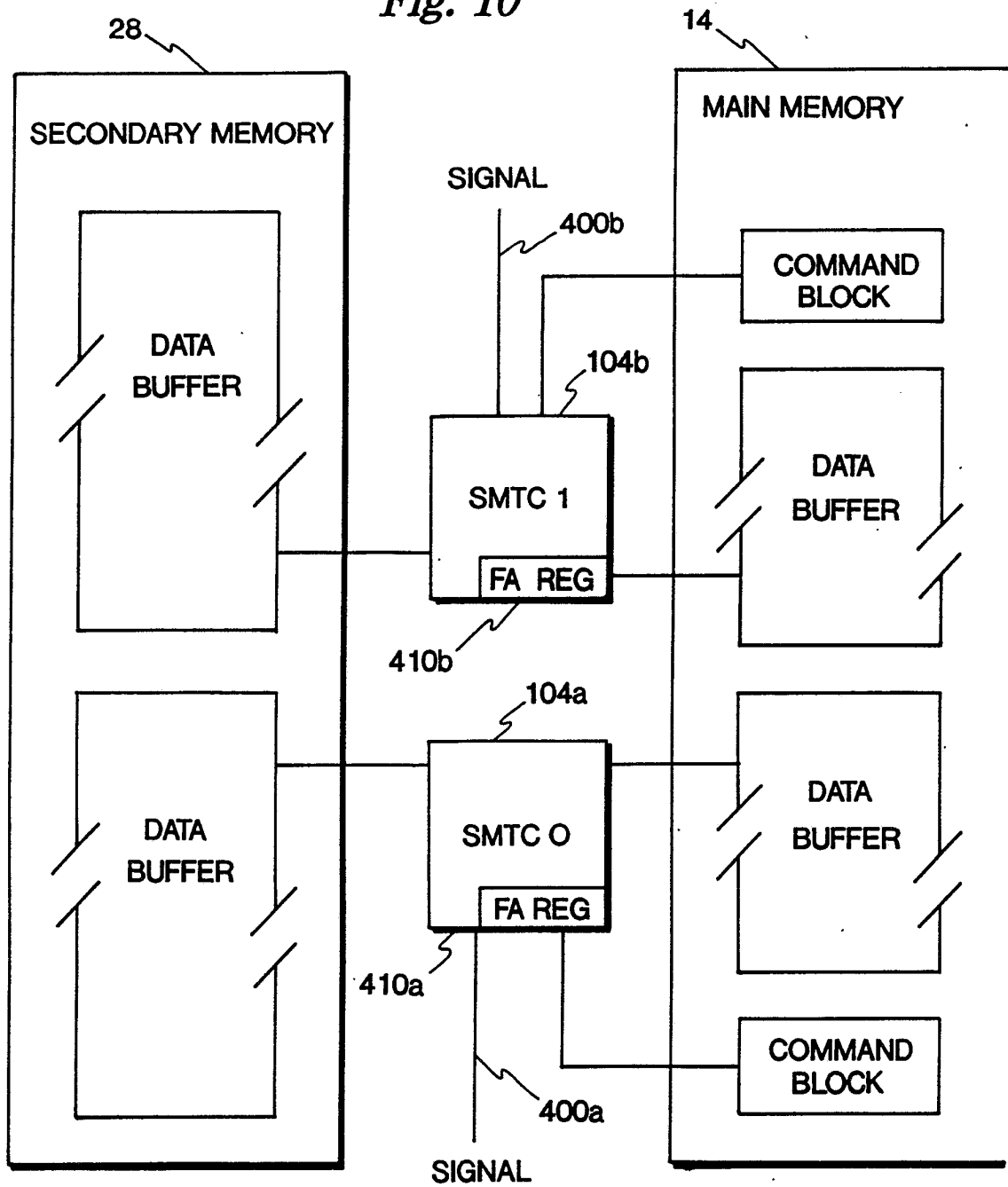
*Fig. 9*

2 <sup>63</sup>	TRANSFER STATUS	2 <sup>0</sup>	WORD 0
	SMTC COMMAND		WORD 1
	SMS FBA		WORD 2
	SMS NBLOCKS		WORD 3
	SMS BLKINCR		WORD 4
	MM FWA		WORD 5
	SIGNAL DEVICE SELECTION		WORD 6

*Fig. 11a*

CODE	COMMAND
0 0 0 0	NOP
0 0 0 1	STORE DATA BLOCK TO SMS
0 0 1 0	NOP
0 0 1 1	NOP
0 1 0 0	NOP
0 1 0 1	NOP
0 1 1 0	NOP
0 1 1 1	NOP
1 0 0 0	NOP
1 0 0 1	NOP
1 0 1 0	NOP
1 0 1 1	NOP
1 1 0 0	NOP
1 1 0 1	STORE ERROR REGISTERS
1 1 1 0	NOP
1 1 1 1	RESET CHANNEL

Fig. 10





13/13

*Fig. 11b*

CODE	COMMAND
0 0 0 0	NOP
0 0 0 1	NOP
0 0 1 0	FETCH DATA BLOCK FROM SMS
0 0 1 1	NOP
0 1 0 0	NOP
0 1 0 1	FETCH ERROR REGISTERS
0 1 1 0	NOP
0 1 1 1	NOP
1 0 0 0	NOP
1 0 0 1	NOP
1 0 1 0	NOP
1 0 1 1	NOP
1 1 0 0	NOP
1 1 0 1	NOP
1 1 1 0	NOP
1 1 1 1	RESET CHANNEL

**SUBSTITUTE SHEET**

# INTERNATIONAL SEARCH REPORT

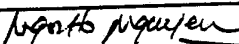
International Application No.

PCT/US91/04060

<b>I. CLASSIFICATION OF SUBJECT MATTER</b> (if several classification symbols apply, indicate all) <sup>6</sup>	
According to International Patent Classification (IPC) or to both National Classification and IPC IPC(5): G06F 15/16 US CL : 364/200	
<b>II. FIELDS SEARCHED</b>	
Minimum Documentation Searched <sup>7</sup>	
Classification System	Classification Symbols
US CL	364/200,900
Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched <sup>8</sup>	

<b>III. DOCUMENTS CONSIDERED TO BE RELEVANT</b> <sup>9</sup>		
Category <sup>*</sup>	Citation of Document, <sup>11</sup> with indication, where appropriate, of the relevant passages <sup>12</sup>	Relevant to Claim No. <sup>13</sup>
Y	US,A 4,484,270 (QUERNEEMOEN) 20 November 1984 (See Figures 1-3).	1-4, 6-10
Y	US,A 4,044,333 (AUSPURG) 23 August 1977 (See Figure 1).	7,9
Y	US,A 4,694,396 (WEISSHAAR) 15 September 1987 (See Figures 1 and 2)	7,9
Y	US,A 3,887,902 (LABALME) 03 June 1975 (See abstract)	5
Y	US,A 3,767,861 (RUTH) 11 July 1972 (See abstract)	2
Y	US,A 4,418,382 (LARSON) 24 November 1983 (See Figures 1-2)	1-4,6-10

<p><sup>*</sup> Special categories of cited documents: <sup>10</sup></p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier document but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p>	<p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.</p> <p>"&amp;" document member of the same patent family</p>
---	---

<b>IV. CERTIFICATION</b>	
Date of the Actual Completion of the International Search	Date of Mailing of this International Search Report
26 July 1991	<b>24 SEP 1991</b>
International Searching Authority	Signature of Authorized Officer
ISA/US	 NGUYEN HOC-HO INTERNATIONAL DIVISION