PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : **G06F 13/40, 13/16**	**A1**	**(11) International Publication Number:** **WO 96/30842** **(43) International Publication Date:** 3 October 1996 (03.10.96)

(21) International Application Number: PCT/EP95/01140

(22) International Filing Date: 27 March 1995 (27.03.95)

(71) Applicant *(for all designated States except US)*: INTERNATIONAL BUSINESS MACHINES CORPORATION [US/US]; Old Orchard Road, Armonk, NY 10504 (US).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: GETZLAFF, Klaus-Jörg [DE/DE]; Friesenweg 26, D-71101 Schönaich (DE). WILLE, Udo [DE/DE]; Wengersteige 47, D-71088 Holzgerlingen (DE). TAST, Hans-Werner [DE/DE]; Hartmannstrasse 66, D-71093 Weil im Schönbuch (DE). LEPPLA, Bernd [DE/DE]; Königstrasse 93, D-71139 Ehningen (DE).

(74) Agent: SCHÄFER, Wolfgang; IBM Deutschland Informationssysteme GmbH, Patentwesen und Urheberrecht, D-70548 Stuttgart (DE).

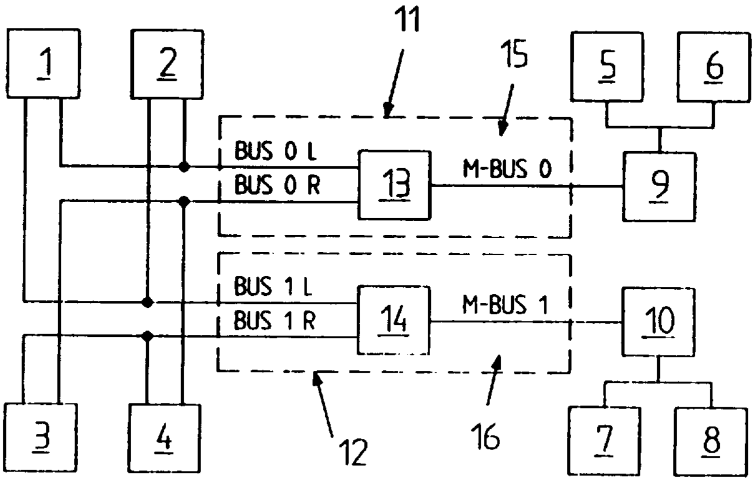
(81) Designated States: JP, US, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: BUS STRUCTURE FOR A MULTIPROCESSOR SYSTEM

(57) Abstract

The invention concerns an improved multiple bus concept for multiprocessor computer systems with a multiple level cache structure. This concept particularly provides an improved bus switching unit. The invention also relates to handling of near end signal reception problems in case of more processors electrically interconnected via such a bus system. The proposed bus structure comprises logical busses each of them being built up by a number of physical bus means. Each logical bus is connected to a memory unit via one or more physical memory bus means and to a bus switching unit (BSU). Each of the processing units is connected to each memory unit via one or more physical busses and via a BSU.



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.

AM	Armenia	GB	United Kingdom	MW	Malawi
AT	Austria	GE	Georgia	MX	Mexico
AU	Australia	GN	Guinea	NE	Niger
BB	Barbados	GR	Greece	NL	Netherlands
BE	Belgium	HU	Hungary	NO	Norway
BF	Burkina Faso	IE	Ireland	NZ	New Zealand
BG	Bulgaria	IT	Italy	PL	Poland
BJ	Benin	JP	Japan	PT	Portugal
BR	Brazil	KE	Kenya	RO	Romania
BY	Belarus	KG	Kyrgystan	RU	Russian Federation
CA	Canada	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic			SE	Sweden
CG	Congo	KR	Republic of Korea	SG	Singapore
CH	Switzerland	KZ	Kazakhstan	SI	Slovenia
CI	Côte d'Ivoire	LI	Liechtenstein	SK	Slovakia
CM	Cameroon	LK	Sri Lanka	SN	Senegal
CN	China	LR	Liberia	SZ	Swaziland
CS	Czechoslovakia	LT	Lithuania	TD	Chad
CZ	Czech Republic	LU	Luxembourg	TG	Togo
DE	Germany	LV	Latvia	TJ	Tajikistan
DK	Denmark	MC	Monaco	TT	Trinidad and Tobago
EE	Estonia	MD	Republic of Moldova	UA	Ukraine
ES	Spain	MG	Madagascar	UG	Uganda
FI	Finland	ML	Mali	US	United States of America
FR	France	MN	Mongolia	UZ	Uzbekistan
GA	Gabon	MR	Mauritania	VN	Viet Nam

DESCRIPTION

BUS STRUCTURE FOR A MULTIPROCESSOR SYSTEM

The invention is concerned with an improved multiple bus concept for multiprocessor computer systems, which predominantly comprise a multiple level cache structure. In particular this concept provides an improved bus switching unit and a method for near end signal reception in case of more processors electrically interconnected via such a bus system.

The bus structure in a multiprocessor computer system has a significant impact on the overall performance, the functionality and the granularity of the system. One of the most important key characteristics for an efficient bus is its bandwidth which describes the ability of the bus to match with the processors speed and a variable number of processors. It depends on the bus width, the efficient usage of the bus by the bus protocol and the bus cycle time.

The bus width is limited by physical constraints of the package as there are chip pads, module pins or wireability of cards and boards. In prior art bus systems therefore either one large or a set of smaller busses is constructed.

In another prior art approaches a smart bus protocol allows bus interleaving between processors to fill the time gap on the bus caused by the access time of the addressed memory bank. "Bus interleaving" is a known method to use wait cycles on a bus to issue a command to other memory banks. It allows several memory banks per bus to be operated simultaneously in order to increase the bus- and memory bandwidth.

The bus cycle time is the main contributor to the bus bandwidth. An efficient bus structure has to provide excellent electrical characteristics to match the processor and the bus cycle time as close as possible. For example, a simple structure which connects processors like a laundry line limits the cycle time and the number of processors by its poor electrical characteristics. In contrast, a much more complex structure provides point to point busses between each processor and a central switch unit. One or several memory banks are connected to that unit. That structure allows a short bus cycle, but requires an extensive wiring. For the packaging, a central switch unit also requires a very high pin count, since all busses are routed to the switch.

In addition, in prior art systems the performance of bus structures is improved by concepts for bus arbitration which may either be implemented as a distributed or a central arbitration function. The central arbitration concept is based on an additional hardware component, a controller, like a central bus switch. It receives all bus requests, performs the arbitration and delivers the grant to the requesting unit in the next cycle.

The distributed approach requires more wiring, because all request lines have to be wired to all bus participants. But its advantage is that the arbitration can be fulfilled within one cycle, since only one off-chip network is in the path, compared with the a central arbitration. The complete path consists of an off-chip network, i.e. the request lines, and the arbitration logic. If this limits the bus cycle time, then a two cycle central bus arbitration will become the preferred solution.

A further key characteristic for an efficient bus structure is the provisions which provides for data consistency assurance in a multiple cache processor structure. A known simple concept is the so called bus snooping. Hereby each processor monitors the bus operations of all other processors to keep track on the status of its cache lines. The status may be "modified", "exclusive", "shared" or "invalid". The required actions are a line invalidation or a cast-out of modified data with or without updating the memory. A cast-out of modified data means that a PU is no longer the exclusive owner of the line. All those operations are initiated and controlled by the individual processors.

But this simple concept has its limits, if there is more than one bus in the structure, as in a system with a central switch. Hereby data consistency has to be controlled by the central switch. It keeps copies of all cache directories of the processors, as well as the status of each line. The central switch monitors the status of the cache lines, initiates cast-outs from processors, if required, and issues the necessary memory commands and the data transfer. Advantageous are the short cycle time and the high bus/memory bandwidth. But a disadvantage of this approach is that it requires a design effort comparable to a complete processor, including a cache. Therefore, this costly concept can be only implemented in high end mainframe designs.

Such a central arbitration concept is disclosed in IBM Technical Disclosure Bulletin, vol. 35, no. 6, Nov. 1992 (Fig. 2) employing a central crossbar switch. Herein, memory requests are transferred via a processor bus, and data via the crossbar switch to a memory unit. This dividing up concept allows a central memory control for an interleaving access to a number of memory banks.

Another approach for central arbitration is given in US-Patent No. 5,355,455 to the present assignee which concerns with the problem of deadlocks in computer systems with several busses being connected by a bus adapter. The bus adapter divides the common bus system into two partial busses wherein the first part contains a processing unit and a memory unit, and the second part contains several I/O control units (Fig. 1 of that document). The underlying problem of that invention is concerned with computer systems having two data busses, where a bus unit of the first bus wants to communicate with a bus unit of the second bus, while a bus unit of the second bus wants to communicate with another bus unit of the first bus, normally leading to a deadlock situation. This problem originates in a bottleneck situation on grounds of the only one bus adapter and is particularly solved by introducing a BUS SUSPEND control signal.

An example for a distributed arbitration approach is disclosed in US-Patent No. 5,093,826 assigned to Siemens Aktiengesellschaft, Germany, which is related to a redundant bus system which is split into two arbitrary halves which are connected to corresponding halves of a redundant main memory. Hereby identical information are stored in the two different memories. According to the teaching of that patent, in a special operating time, individual processors, one bus system half, and some of the memory sections are split off and interconnected to form an independent special-purpose computer.

European Patent Application No. 0 557 651 concerns with a multiprocessor system which allows an efficient interleaving on a memory bus. This concept allows multiple access of bus participants to the memory since the memory is logically divided into at least two memory banks.

Further, in German Patent GE-PS 3 708 887, a parallel bus structure is disclosed, whereby the bus is divided into sections by means of registers being inserted between those sections. In each bus section at a given time data are transferred independently of each other, or data are pipelined. Due to subdividing the data bus in individual bus sections with a physical length of typically centimeter, the signal transmission time is strongly reduced. Therefore higher cycle times and transfer rates are achievable.

It is therefore an object of the present invention to provide an improved bus concept for a tightly coupled multiprocessor computer system which aims at maximum bus bandwidth by the usage of a simple and efficient multiple bus system at minimal bus cycle time.

It is another object to provide a bus structure which comprises a highest amount of system's modularity with respect to the extension and the reduction of the computer system where this bus concept is implemented.

A further object is to provide a bus structure which aims at maximum point to point connectivity between the bus participants.

The bus structure according to the present invention solves this problem by providing logical busses, which are built by two or more physical busses. The number of physical busses is variable and may scale with the number of processors interconnected by that bus system. Further, each logical bus contains a bus switching unit (BSU) and is connected to a memory bank, wherein the processors are connected to each logical bus via physical busses and the bus switching unit(s).

The physical busses allow each processor to have access to each logical bus. Hereby parallel access of one or more processor units to one or more memory units is supported in a very efficient manner. It is accentuated that this parallel processing concerns the access of one processing unit to several memory units as well as the access of several processing units to several memory units.

Moreover, data consistency is accomplished by a bus arbitration logic which allows only one processor access to a logical bus at a given time. This automatic serialization of bus requests for the same line by several processors simply allows exclusive ownership by only one processor at a given time. Additionally, bus interleaving for different memory units on a logical bus can be simply fulfilled.

With this multiple logical bus concept a particular connectivity and bandwidth improvement over a single bus concept with double or four times bus width is achieved.

Linefetch operations on a bus generally consist of command-, memory wait-, and data transfer cycles. A single bus improvement does only reduce the number of cycles for a data transfer, but not for command or memory wait operations. This multiple logical bus concept requires less cycles per bus operation, because the complete operations can be fulfilled concurrently. The loss of one cycle during the BSU operations has only a small impact on the systems performance with respect to the smaller cycle time.

The BSU in the proposed bus structure serves as a switch 1>n by which command/control signals and data from one port are driven to all other ports in the next cycle. Each BSU works only on one logical bus which is split up in several physical busses to improve the bus cycle time due to the electrical point to point connections between the different bus participants. The logical busses are routed all the way via the respective BSUs to their memory units. So this set-up allows a parallel selection of the memory units within the system.

By splitting a second-level cache memory unit into at least two parts it is accomplished that each single processor or a group of processors have an electrical point to point connection to each of the BSUs. Thus the physical structure of these BSU busses can be built such, that it allows a bus cycle equal or close to the processor cycle.

Further the variable number of physical busses allows simple adaption of the bus structure to processor and bandwidth requirements, e.g. in case of an extension of the number or group of numbers of processors.

In particular, due to its modularity, the proposed structure can be used to build a variety of systems which are optimized towards high performance or cost/performance or low cost. Different packages may be chosen like multiple chip modules for high end systems with a large number of logical busses and processors or as a low end system with two processing units and one logical bus, mounted on single chip modules on card.

Further this modular system structure allows system operation after hardware defects in a degraded mode, e.g. with a reduced number of processing units, a reduced number of processor clusters or a reduced number of busses or memory cards, respectively.

The proposed bus concept can not comprise separate command-, address- and data busses. For limitation of the signal count they are all multiplexed on the same physical busses. Therefore a different bus communication is required, depending on command, address or data on the bus. For the details of this communication it is referred to the figurative description.

In another embodiment, these communication protocols are handled by the BSU, particularly including the redrive mechanism for simple broadcasting information on the overall bus.

In a further preferred embodiment of the invention, data consistency is further improved by providing a bus arbitration unit for granting access to a logical bus to only one processor at a given time. The serialization of bus requests for the same line by several processors is enforced by assigning the logical busses to address classes. Assigning each logical bus to only one address class assures data consistency very efficiently. Hereby each bus can have its own congruence class, determined by e.g. the low order line address bits. A requesting processor selects the proper logical bus only by this criterium. The overall performance of the bus structure can be further improved by implementing the bus arbitration function within the bus switching unit. The cycle time of the bus can be further improved by a redrive logic on the bus toward the command and the data direction, e.g. Pu>Cache>Bus>Redrive>Bus>Memory (Command) or

Memory>Bus>Redrive>Bus>Cache>Pu (Data). Those partial busses allow shorter cycle times, however one cycle has to be spent in the redrive logic in each direction.

The BSU connects different physical busses between a number of processor units, cache memories, memory bus adapters and storage controllers, and redrives incoming data to those bus participants. Thus it may involve the bus redrive and control signal redrive functions, whereby the BSU does not have to contain any functions to assure data consistency among the processor caches. It rather redrives the command cycle of an active processor from one bus back to all the other busses including the command select signal. This causes the other processors to perform a bus snooping operation for the line in question.

By the proposed structure all bus snooping actions can also remain in the processors themselves. No copies of the cache directories have to be maintained. Further, no bus cycle is lost by the redrive of a command on the other busses, because the BSU issues commands on the memory bus concurrently.

The bus switching unit further serves as a memory interface, which also matches the processor cycle time, since this is a point to point connection between BSU and memory.

It is further important to contain the associativity of the data lines throughout the complete memory hierarchy. Advantageously all line replacement actions, as they come up during a linefetch operation, are always related to the initially selected logical bus and the memory unit. There is no interference between busses nor interleaved lines.

The redrive function can be also fulfilled exclusively by the processors. The memory units preferably comprise at least two memory banks being controlled by a storage controller. The logical busses are routed all the way via the respective BSU's to their memory banks which allows a parallel selection of memory banks within the system. All line replacement actions, as they come up during a linefetch, are always related to the initially selected bus and bank. There is no interference between busses nor between interleaved lines.

Another aspect of the invention is a method for handling near end signal reception problems in the proposed bus structure where clusters of processors are not interconnected by point to point connections. In particular, all near end drive/receive operations, like bus snooping or casting-out on the same physical bus, whereby one processor possesses an actual line and has to transfer it to a neighboring processor, are executed in two cycles. In particular, dependent on the actual signal reflection behaviour, it is interchanged dynamically between the one-cycle and the two-cycle operation mode of the information transfer. Further only a one-cycle control signal is needed for the overall bus structure.

These and other objects, features and advantages of this invention will become apparent to those skilled in this art from the following more particular description of one presently preferred embodiment of the invention, taken in conjunction with the accompanying drawings, in which:

Fig. 1 is a block diagram showing the fundamental features of the multiple bus structure according to the invention;

Fig. 2 is a table showing the line assignment of a structure with two logical busses allowing a two-way interleaving on each memory bus;

Fig. 3 is an embodiment of the bus structure according to the present invention in a low end computer system which comprises two processors with a shared L2-cache memory being connected to a memory bank via one logical bus;

Fig. 4 is a further embodiment of the bus structure in an intermediate computer system which comprises six processors being connected to memory banks via two logical busses;

Fig. 5 is a preferred embodiment of the invention in a high end computer system with 16 processors being connected to four memory banks via four logical busses;

Fig. 6 shows a preferred implementation of the BSU (bus switching unit) according to the invention;

Fig. 7 shows a complete control signal structure of the basic bus operations accomplished by the BSU;

Fig.s 8 and 9 are timing charts showing the bus protocol for a "linefetch" and a "linefetch with cast-out operations" on a bus structure according to this invention;

Fig. 10 is a schematic block diagram of a processor cluster connected to a logical bus via a BSU, illustrating signal reception behaviour of a multiple bus due to bus delays (no electrical point to point connections); and

Fig. 11 depicts the bus structure of Fig. 10 together with the respective control signals.

Fig. 1 is a schematic overview of a multiprocessor computer system comprising a multiple bus structure according to the invention. The multiprocessor system exemplary consists of four processors 1, 2, 3, 4 with no private L2 cache memories and a main memory which is divided exemplary into a first memory bank built by two memory modules 5, 6 and a second memory bank built by two memory modules 7, 8. The memory banks 5, 6 and 7, 8 are controlled by storage controllers (STCs) 9, 10.

The common bus system is divided in two logical busses 11, 12 which are labeled by logical reference numbers 0 and 1, respectively, wherein each of the logical busses comprises a bus switching unit (BSU) 13, 14. Each of the processors 1, 2, 3, 4 is electrically (or a light connection within a laser computer) connected to each of the BSUs 13, 14 whereby providing a bus network (no point to point connections) with four physical busses "0L", "0R", "1L", "1R".

The BSUs 13, 14 connect the four physical busses on the processor side to the memory banks 5, 6, 7, 8 via memory busses 15, 16. It is noteworthy that the number of logical busses 11, 12 equals the number of BSUs 13, 14. The BSUs 13, 14 contain bus arbitration functions (not shown) for controlling access to the logical busses 11, 12. It is emphasized that each of the logical busses 11, 12 provides a single line connection to a memory bank 5, 6, 7, 8.

It is further emphasized that, by this multiple bus concept, the number of processors 1, 2, 3, 4 and the number of memory units or memory banks 5, 6, 7, 8, respectively, per logical bus 11, 12 can be extended easily. In case of adding a number of processors, no further BSUs 13, 14 are required. In accordance, improving of the system by adding further memory banks, may change the number of logical busses or BSU's, respectively.

Further key elements of this bus structure are that the logical busses 11, 12 are built by a variable number of physical busses and that both logical busses 11, 12 allow n-way bus interleaving, e.g. a 4-way interleaving in the embodiment according to Fig. 1, 2-way for each logical bus 11, 12 due to the two memory banks 5, 6, 7, 8. But there may be multiple banks per logical bus 11, 12 to support a more complex bus interleaving.

The bus structure according to Fig. 1 does not comprise separate command-, address- and data busses. To limit the signal count they are all multiplexed on the same physical busses (Fig. 6). Therefore, a different bus communication is required, depending on commands, addresses or data on the bus (Fig. 8 and 9). For instance, data are only exchanged between processors 1, 2, 3, 4 and BSUs 13, 14, with the exception of cast-out operations. But command and address cycles require both, processor-BSU and processor-processor communication.

Each of the logical busses 11, 12 in Fig. 1 is assigned to its own congruent address class. This is illustrated with respect to Fig. 2 for the exemplary case of a 2-logical bus system with 2-way interleaving on each logical bus according to Fig. 1. The total memory address space is divided into blocks of cache line size and each of the blocks is associated with a bus and a memory bank by the low order address bits.

It is hereby assumed that eight lines 0-7, e.g. cache lines in case of processors with second-level L2 cache memories, are stored in the main memory 5, 6, 7, 8. Each line is characterized by the number of the logical bus and the memory bank where it is stored.

The congruence class can be easily determined by the low order address bits of the lines. A requesting processor 1, 2, 3, 4 selects a proper logical bus 11, 12 only by this criterium. The logical busses 11, 12 are routed all the way via the respective BSU's 13, 14 to their memory banks 5, 6, 7, 8. This set-up allows a parallel selection of the memory banks 5, 6, 7, 8 within the whole system whereby ensuring that each line can be accessed by only one processor at a given time. If more processors request an access to the same line, due to the serialization of access automatically achieved by these classes assures very effectively data consistency in the whole computer system.

A further advantage of this set-up is, that all line replacement actions, as they come up during e.g. a linefetch operation, are always related to the initially selected logical bus 11, 12 and the initially selected bank 5, 6, 7, 8. There is no interference between busses nor between interleaved lines.

The proposed bus structure can be implemented within the whole range of so-called "low end" and "high end" multiprocessor computer systems comprising at least two processor units. In Fig.s 3 to 5 this broad range of applicability is illustrated by its implementation within three different systems.

At first, in Fig. 3 a low end computer system with two processors 101, 102 is shown, which are connected to a shared L2 cache memory 103 via physical busses 104, 105. The system also provides a main memory unit 106 which exemplary includes two independent operating memory banks 107, 108. A storage controller (STC) is provided in order to control access to the memory banks 107, 108. The processors 101, 102 and the memory units 107, 108 are interconnected by a bus switching unit (BSU) 109 via a logical bus 110. The BSU 109 provides also an input/output port 111 which serves as a connection to external I/O devices. In this structure both processors 101, 102 have an electrical point to point connection to each of the memory banks 107, 108. This system can be easily extended by adding more processors or memory modules. In the latter case for each memory unit only a further logical bus has to be implemented, not changing the existing part of the whole system.

Fig. 4 depicts a system with six processor units 201-206 and two logical busses 207, 208 as an example of an intermediate computer system. Three processor units are connected to shared L2 cache memories 209, 210. Each of the logical busses 207, 208 provides a BSU 211, 212 and a STC 213, 214 and is connected to a memory unit 215, 216 each of them built by two memory banks 217-220, respectively. It is seen, that this bus concept provides electrical point to point connections between the processors 201-206 and the memory banks 217-220. Therefore, each line stored in the main memory can be only accessed via one physical bus.

An implementation of the bus structure in a preferred high end computer system with sixteen processors, and four logical busses 305-308 is shown in Fig. 5 only four processors 301-304 are shown. Further each logical bus 305-308 provides a main memory unit 309-312 whereby the memory units are built by memory banks 313-320. It is noteworthy that each group of four processor units, e.g. group 301-304, is connected to two L2 cache memories 321-328. Despite of the complexity of this system, all processors 301-304 and memory banks 313-320 are linked by a simple point to point connection. Moreover, due to splitting the L2 cache memories in two parts not only two processor units 301-304 but also one processor unit can communicate with the main memory at a given time.

In Fig. 6 a block diagram of the redrive logic of the BSU according to the invention (within a system as in Fig. 4) is depicted in more detail, together with the data flow inside the BSU. The BSU provides four bi-directional ports 401-404 wherein the ports 401-403 represent ports to the bi-directional physical busses on the processor side of the BSU, and the port 404 is connected to a bidirectional memory bus. The incoming traffic first passes a bus receiver 405-408, wherein the control signals of bus drivers 409-412 are disabled. The incoming traffic is transferred to a multiplexor 413 where the appropriate signals are selected by incoming command signals 414. The traffic is further buffered in registers 415-418 and thereafter fed into the drivers 409-412. From here the signals (traffic) are redriven to all busses via ports 401-404. Signals produced internal by the BSU, itself, are transferred via a bus receiver 419 to the multiplexor 413.

Thus the BSUs serve as switches 1>n which connect and redrive command/control signals and data from one port to all the other ports in the next cycle. In a further embodiment, the BSU contains different functions, namely a bus redrive and a control signal redrive function, a memory interface and bus arbitration. The BSUs redrive commands and data from one source port to all other ports, including the memory bus. Most of the respective control signals are also redriven. Complete informations are buffered in the BSUs and distributed in the following cycle.

Hence each single processor or a group of processors has a point to point bus connection to each of the BSUs, a bus cycle equal or close to the processor cycle is achieved. All near end drive/receive operations, like bus snooping on the same physical bus, are accomplished in 2 cycles. Herethrough it is guaranteed that electrical signal reflections at the end nodes of the physical bus structure are died down at least in the second cycle. Thus reflection noise signals can be suppressed very efficiently.

The point to point busses allow optimal short bus cycles and can be implemented by grouping of processors and L2 cache chips in building blocks (clusters). This structure can be built by point to point busses between processors and L2 cache chips as well as between each group and their respective BSU chip(s).

The memory interface between the BSUs and the STCs consists of only four signals: A Xfer data signal which indicates that data are on the bus, an Allow transfer data which confirms a data request by a processor, a Command select which indicates that a command is on the bus, and a Memory error, which indicates an error or invalid data on the bus.

Bus arbitration can be also performed by the BSUs. Hereby all requests are collected and honored with a grant signal to the requesting processor when allowed by the arbitor logic and when the bus is available.

The BSUs do not comprise any functions to assure data consistency among the processor caches. They rather redrive the command cycles of active processors from one bus back to all others, including the Command select signal. This causes the other processors to perform a bus snooping operation. If one or more processors find the line, they raise a busy signal to the BSUs, until they have taken the required action. Again, the busy signals are redriven to all other processors, including the requesting one. The implementation of the bus snooping functions in the processors are only small add-on's, because most of the required functions are already contained in the cache control unit of the processors.

The basic bus operations accomplished by the BSUs are illustrated in the following with regard to Fig. 7:

* PU REQUEST
This control signal concerns a processor bus request to the BSU arbitration logic. A request can be principally directed to e.g. linefetch/linestore, line invalidate or I/O fetch and store operations. The requesting PUs decide by means of the address classes, on which logical bus a request has to be transferred. Herethrough a definite access to a memory bank can be achieved.

* PU CMD SELECT
The control signal being sent by a PU indicates that a command, a field length and an address are on the bus. The signal is received by a definite BSU which redrives the signal in the following cycle on all other busses, including the memory interface as a BSU CMD SELECT.

* XFER DATA
This signal being sent by the memory indicates that data are on the bus, e.g. due to a linefetch operation. The signal is received by a definite BSU which redrives the signal in the following cycle on all other busses as an BSU XFER DATA signal. In case of a cast-out operation, the signal is generated by a BSU.

* PU BUSY

This signal is a response from a processor indicating that a matching line is found. The processor keeps the signal on until the required action is completed. The signal is received by a definite BSU and redriven in the following cycle on all other busses as a BSU BUSY signal.

* BSU GRANT
A BSU grants its logical bus to a requesting PU as a result of an arbitration which results in an availability of that bus.

* ALLOW XFER DATA
This control signal confirms a data fetch/store command to the memory and indicates, that no cast-out will occur. The signal is generated by a BSU when the signal PU BUSY is not raised or dropped without a cast-out as a result of bus snooping.

* MEMORY ERROR
This error indicating signal is generated by the memory controller of a memory bank. The signal is received by the BSU and redriven in the following cycle on all other busses as a BSU MEMORY ERROR signal.

All those signals are provided for all physical busses. Their total amount may differ with the number of processors, connected to one bus.

The bus structure and the bus protocol allow synchronous bus snooping for all processors without relevant performance degradation. This avoids cache directory copies of all processors in the BSU and finally allows multiplexing of addresses and data on the same bus, wherein no extra address bus is required.

In Fig. 8 the functionality of the bus structure according to the invention is illustrated by means of a timing chart representing a linefetch bus operation. In order to simplify the basic aspects of the bus concept, the operations relate to a system embodiment comprising only two processors (PU^0, PU^1) which communicate via a logical bus being splitted up in two physical busses (0L, 0R) which are connected to a bus switching unit (BSU). Further the BSU is connected to a memory unit via a memory bus (M-BUS).

First PU^0 sends a linefetch request to the BSU. After an arbitration cycle (A) the BSU grants the logical bus to the requesting PU^0 as a result of its availability. PU^0 generates a PU CMD SELECT control signal which indicates that a command, a linefetch address and a length (C) are on physical bus 0L. The BSU redrives the PU CMD SELECT signal in the following cycle on the other physical busses, i.e. BUS "0R" and M-BUS, as a BSU CMD SELECT signal. Therefore, the command, linefetch address and the length information reach those busses with one cycle latency.

The BSU CMD SELECT signal on bus 0R causes PU^1 to perform a bus snooping operation within a bus snooping time window. PU^1 does not find the requested line and so the PU BUSY control signal becomes not active. This control signal would only become active if PU^1 would find a matching line. In case of other processors connected to physical bus 0L, also a PU BUSY signal on physical bus 0L would have to be regarded. With the PU BUSY signal being not active, the BSU confirms the linefetch command to the memory unit by generating a BSU ALLOW XFER DATA control signal.

The memory unit now sends the requested data to the BSU via the M-BUS wherein an XFER DATA control signal is generated by the memory in parallel. This control signal and the data are received by the BSU and redriven with one cycle latency on all other busses. The control signal especially is redriven as a BSU XFER DATA control signal. Both control signals keep on the busses as long as data are on the busses.

Fig. 9 shows, for example, a timing chart for a cast-out operation in response to successful (matching) bus snooping in a system according to Fig. 8. In particular, herein the near end reception problem is illustrated.

Again, in the first place PU^0 requests for a line (linefetch) via physical bus 0L. The BSU grants the bus to PU^0 and therefore a command, a linefetch address and a length get on the bus 0L of that PU. Further a PU CMD SELECT control signal is generated in parallel by the PU indicating this state of the bus. This control signal is received by the BSU and redriven in the following cycle on all busses as a BSU CMD SELECT signal. This signal initiates a bus snooping operation of PU^1. It is supposed that PU^1 finds the line and raises a PU BUSY control signal to the BSU. The signal is kept on the bus until the required action of PU^1 is completed, and is received by the BSU and redriven in the following cycle on all other busses as a BSU BUSY control signal.

PU^1 puts a cast-out command on bus 0R which is redriven with one cycle latency to all other busses. At the same time PU^1 raises a PU CMD SELECT signal to the BSU which is redriven to all other busses in the following cycle. Now the eight data packets are transferred from the memory unit to PU^1.

This processor-processor communication on that near end leads to an electrical signal reflection behavior wherein the signals which are raised by the both processors and redriven by the BSU interfere on each physical bus. The signal reflection of the present bus structure can be used to optimize the bus cycle time. The bus delay depends on whether a signal is sent to a near-end or a far-end receiver, since the reflection returning from the far-end disturbs the reception on the near-end side. That phenomenon is well-known for all non-terminated physical networks. Fig. 10 exemplary shows this type of a physical bus. Hereby the proposal of the invention is to keep a physical bus being disturbed for two cycles valid for commands, bus snooping addresses and cast-out data in order to allow near-end reception. Due to the cast-out operation, a BSU XFER DATA control signal is generated by the

BSU for the second cycle of each data block on bus 0R. The data blocks are also redriven to the other busses with a one cycle latency. This one cycle delay of the control signal has the advantage that those reflected signals have only a small impact on the systems performance but relaxes these critical bus operations. Fig. 11 shows these busses together with the respective control signals. Hereby CMD SELECT and XFER DATA signals are driven from the BSU indicating the bus validity on both, the originating and the redriving bus.

CLAIMS

1. Bus structure for a multiprocessor system comprising at least two processing units, characterized by

   at least one logical bus which is set up by at least two physical busses,

   wherein each logical bus provides a connection to a memory unit and to a bus switching unit, and

   wherein the processing units and the memory units are interconnected by the bus switching units via the physical busses.

2. The bus structure according to claim 1, wherein addresses, commands and data are transferred via single lines of the physical busses.

3. The bus structure according to claim 1 and/or 2, wherein each logical bus comprises a bus arbitration unit which grants access to a logical bus to only one processing unit at a given time.

4. The bus structure according to one or more of claims 1 to 3, wherein logical busses are assigned to at least one address class being defined by reference numbers of the logical busses and the memory units, the number of address classes depending on the number of independent memory units.

5. The bus structure according to one or more of the preceding claims, wherein the number of physical busses scales up and down with the number of processing units or groups of processing units.

6. The bus structure according to one or more of the preceding claims, wherein the memory units comprise

at least two memory banks which are controlled by storage controller means and which allow n-way bus interleaving.

7. The bus structure according to one or more of the preceding claims, the multiprocessor system comprising at least two second level cache memory units on a logical bus, wherein each of the processing units is connected to each of the cache memory units by a single line.

8. The bus structure according to one or more of the preceding claims, wherein the bus switching unit comprises:

receiving means and register means for gathering and buffering incoming information;

driving means for driving/redriving information in/out of the bus switching unit;

multiplexing means for selecting the appropriate signal for the redriving; and

redrive logic means for redriving information to all the other ports via the multiplexing means in the next cycle.

9. The bus structure according to claim 8, wherein the bus switching unit further comprises arbitration means for arbitrating the bus after a certain bus protocol.

10. The bus structure according to one or more of the preceding claims, comprising a cluster of at least two processing units connected to a bus switching unit via branched physical busses, wherein near end signal reception is accomplished by dynamically

looking for signals which are transferred between two processing units which intercommunicate via an originating and a redriving physical bus; and keeping these signals for at least two bus cycles on the bus.

11. The bus structure according to claim 10, wherein the bus switching unit provides a one-cycle control signal for near end reception signals.

12. The bus structure according to claim 10 and/or 11, wherein the bus switching unit provides a one-cycle control signal on the second cycle of the signal on the originating bus.

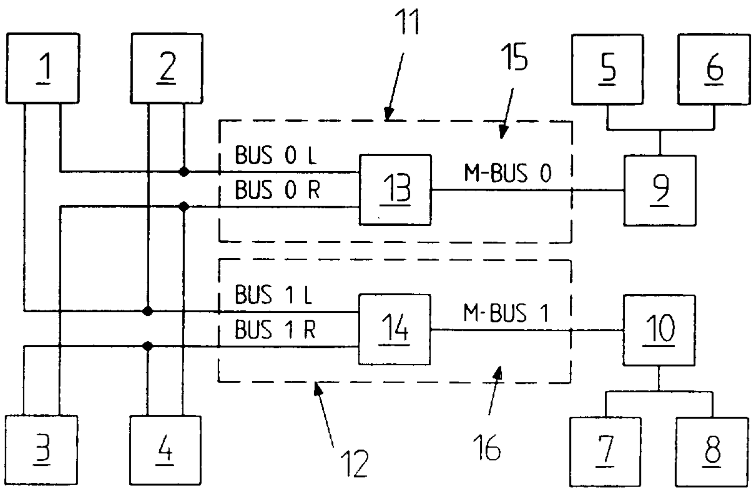
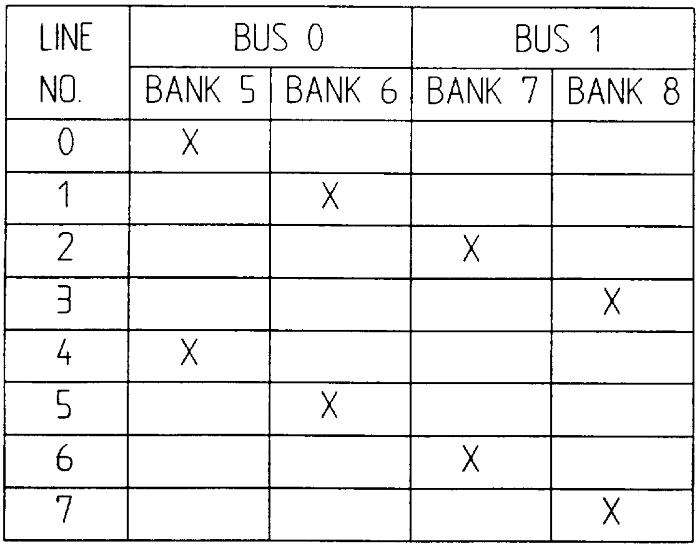


FIG. 1



LINE NO.	BUS 0		BUS 1	
	BANK 5	BANK 6	BANK 7	BANK 8
0	X			
1		X		
2			X	
3				X
4	X			
5		X		
6			X	
7				X

FIG. 2

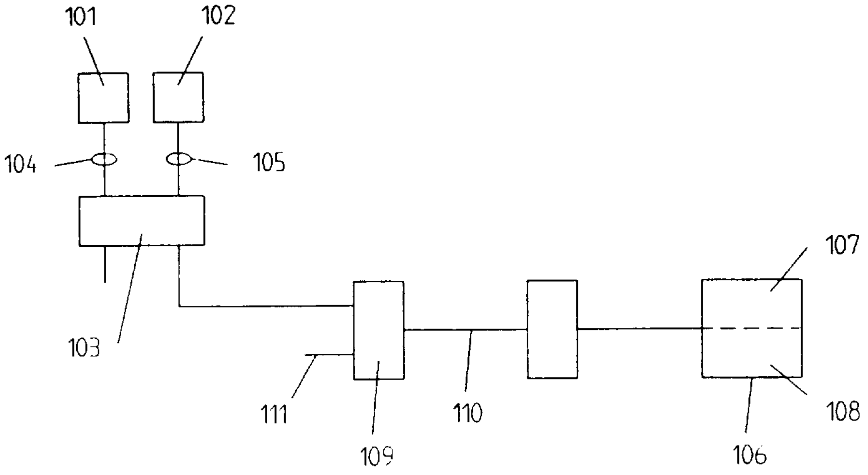

101
102
104
105
107
103
111
110
108
109
106

FIG. 3

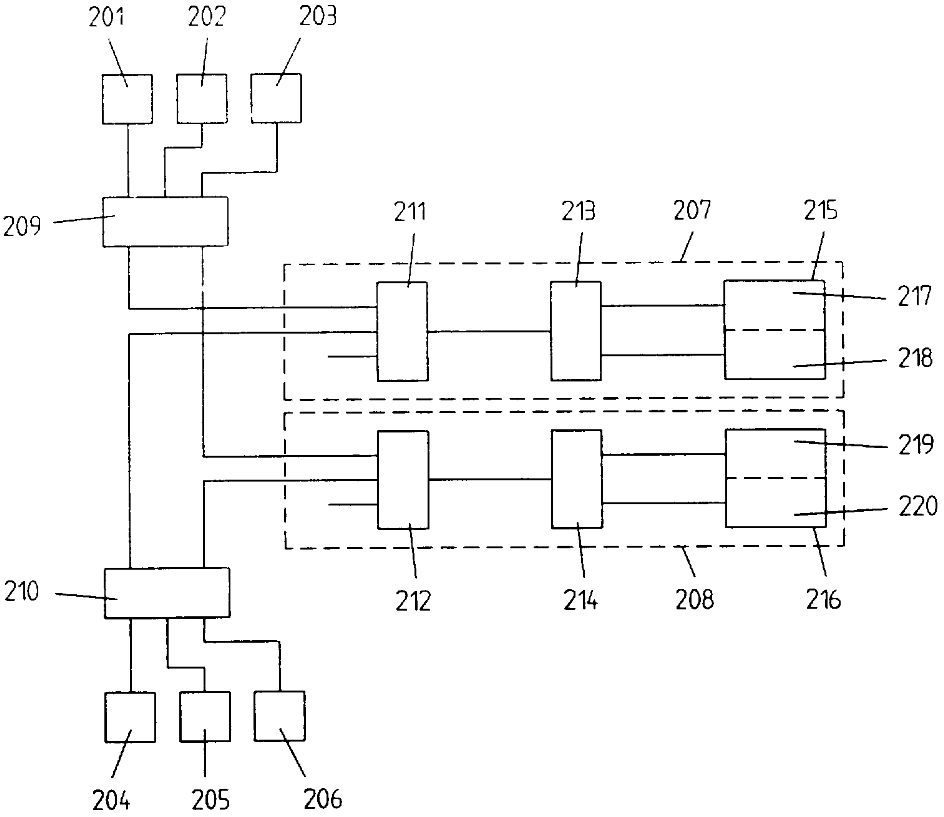

201
202
203
209
211
213
207
215
217
218
219
220
210
212
214
208
216
204
205
206

FIG. 4

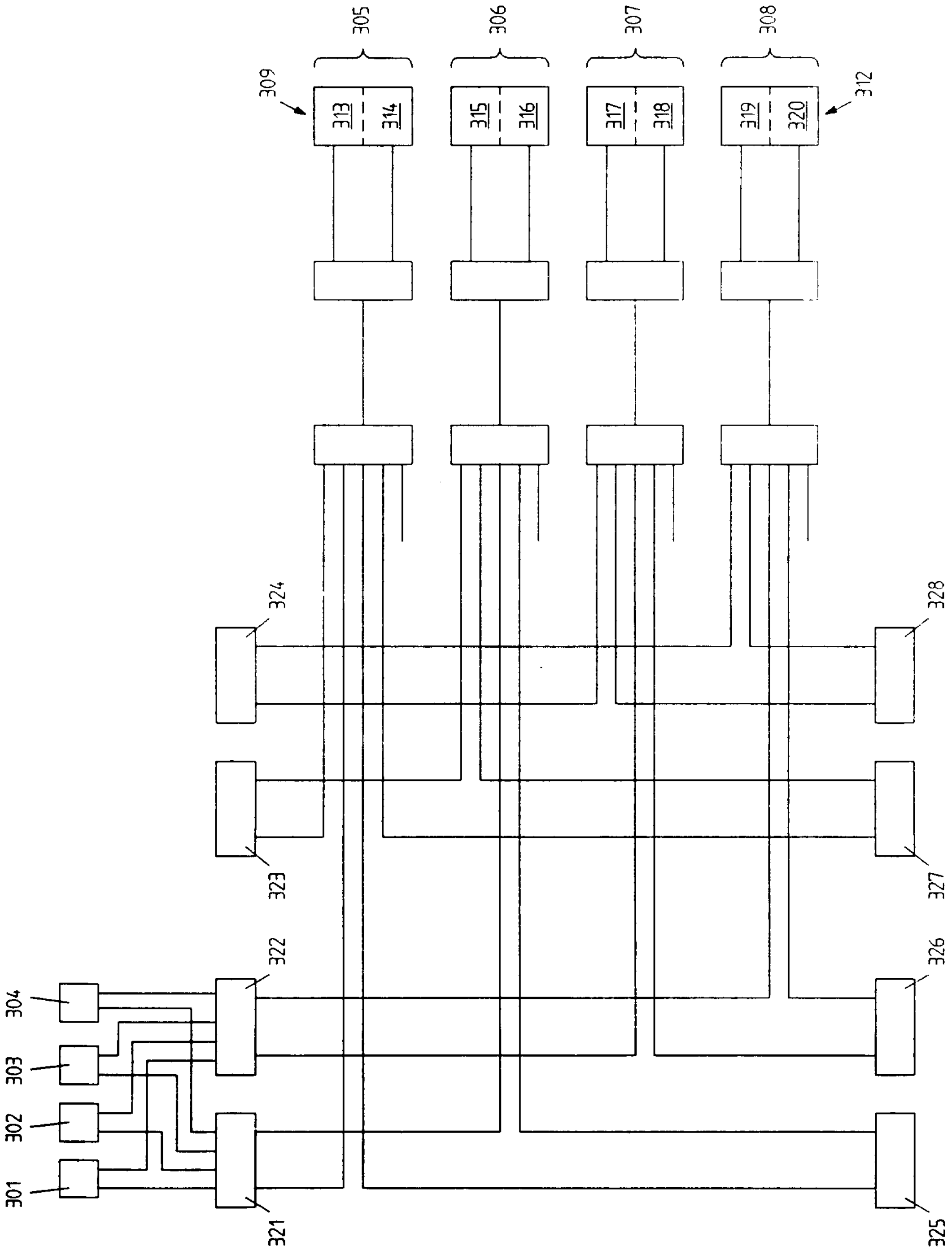

305
306
307
308
309
312
313
314
315
316
317
318
319
320
324
328
323
327
322
326
304
303
302
301
321
325

FIG. 5

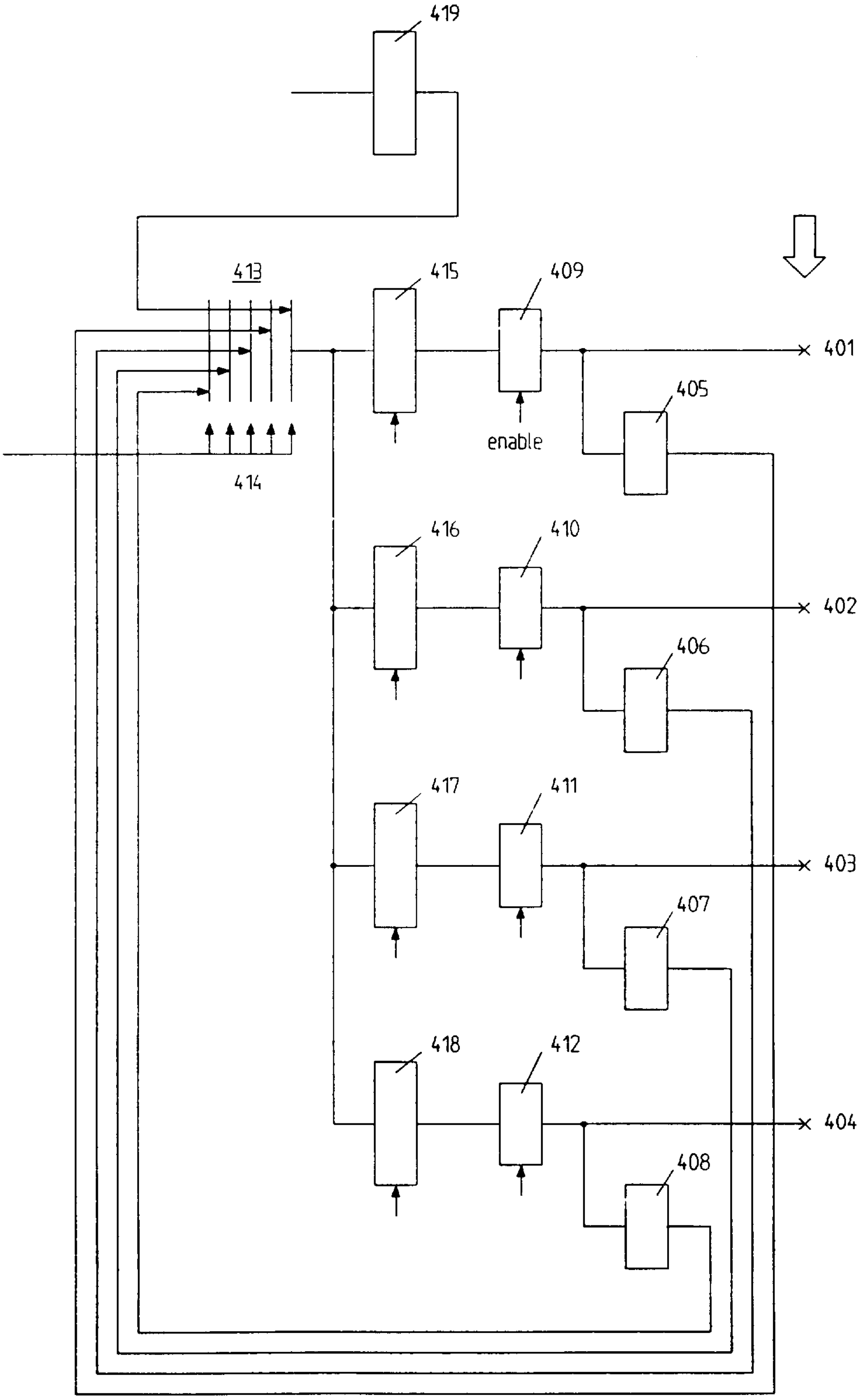

419
413
415
409
401
405
enable
414
416
410
402
406
417
411
403
407
418
412
404
408

FIG. 6

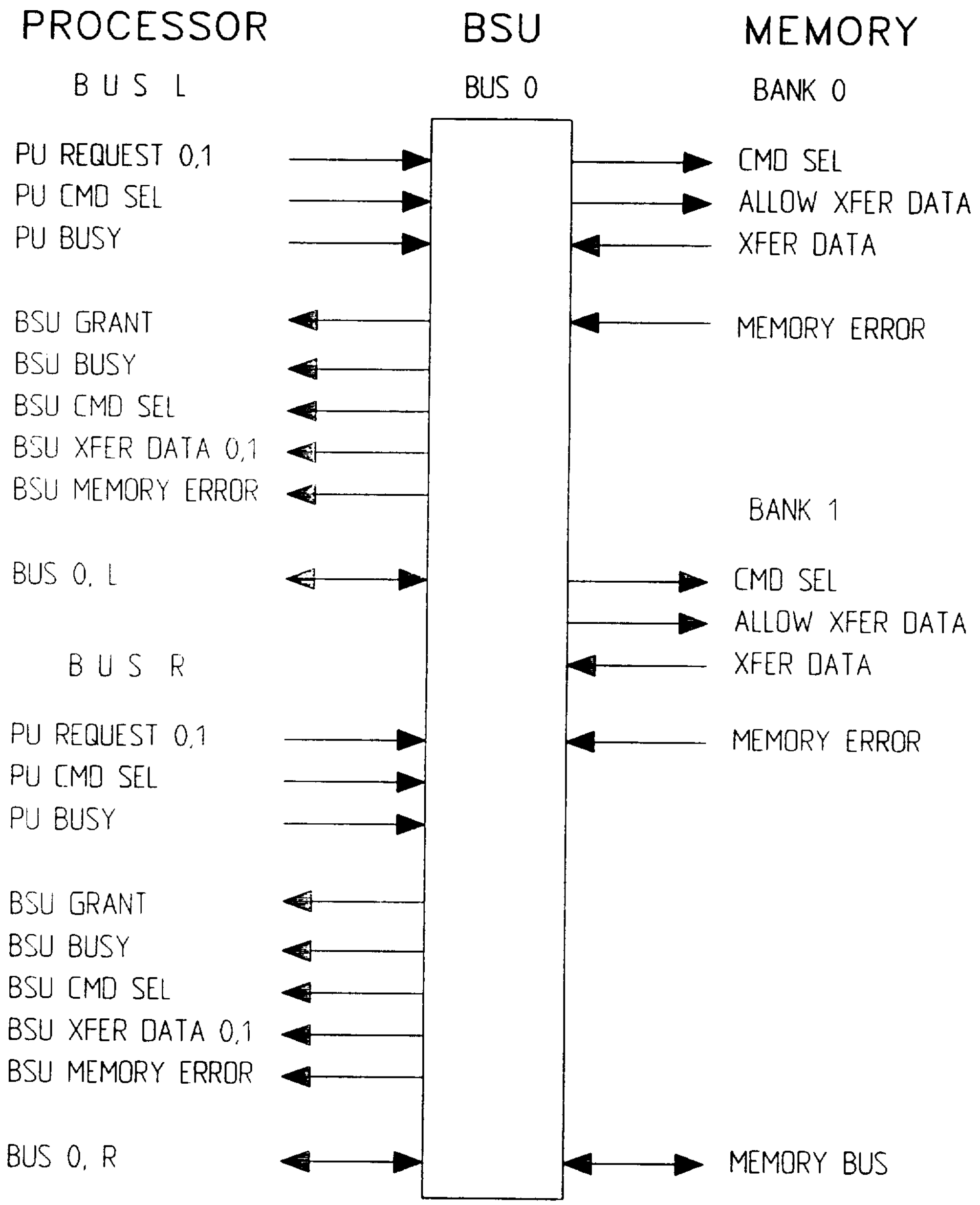
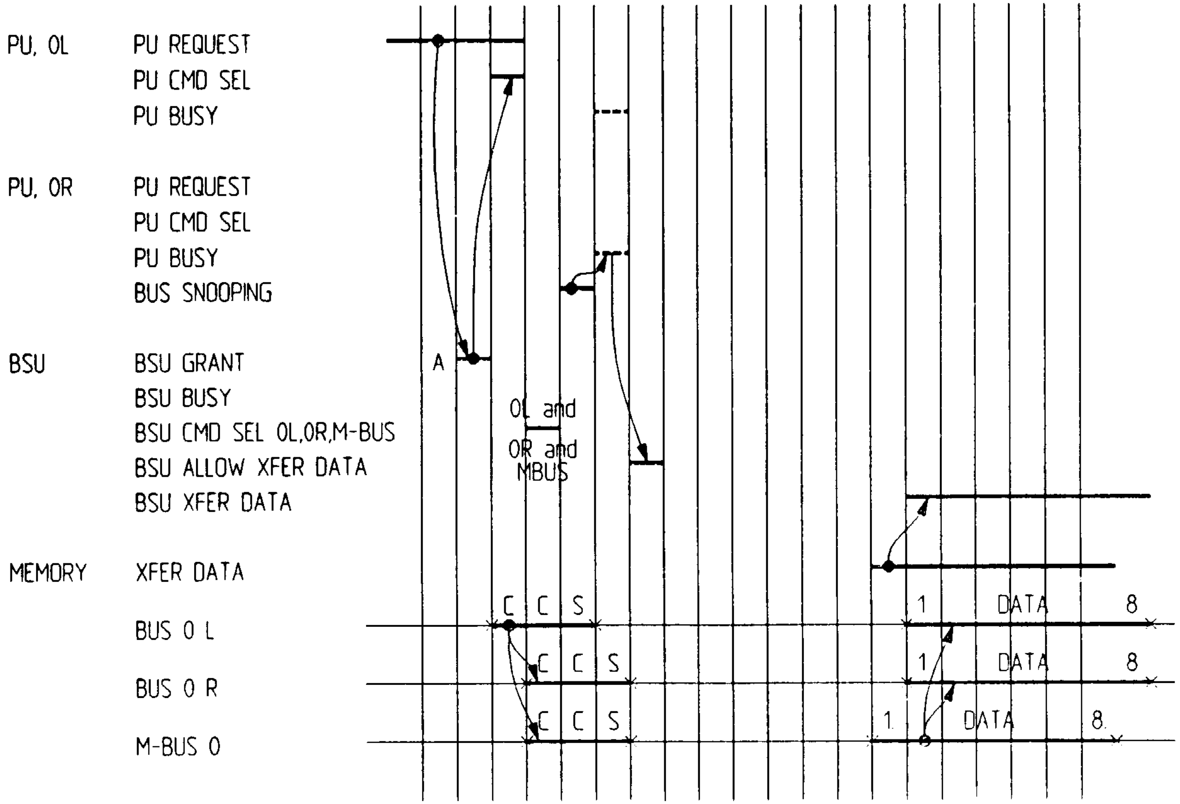

PROCESSOR
BSU
MEMORY
BUS L
BUS 0
BANK 0
PU REQUEST 0,1
PU CMD SEL
PU BUSY
BSU GRANT
BSU BUSY
BSU CMD SEL
BSU XFER DATA 0,1
BSU MEMORY ERROR
BUS 0, L
BUS R
PU REQUEST 0,1
PU CMD SEL
PU BUSY
BSU GRANT
BSU BUSY
BSU CMD SEL
BSU XFER DATA 0,1
BSU MEMORY ERROR
BUS 0, R
CMD SEL
ALLOW XFER DATA
XFER DATA
MEMORY ERROR
BANK 1
CMD SEL
ALLOW XFER DATA
XFER DATA
MEMORY ERROR
MEMORY BUS

FIG. 7

BUS OPERATION / LINEFETCH



A = ARBITRATION CYCLE
C = COMMAND LINE FETCH ADDR, LENGTH
S = LINE STORE ADDR (IF REQ)
---- = NOT ACTIVE

FIG. 8

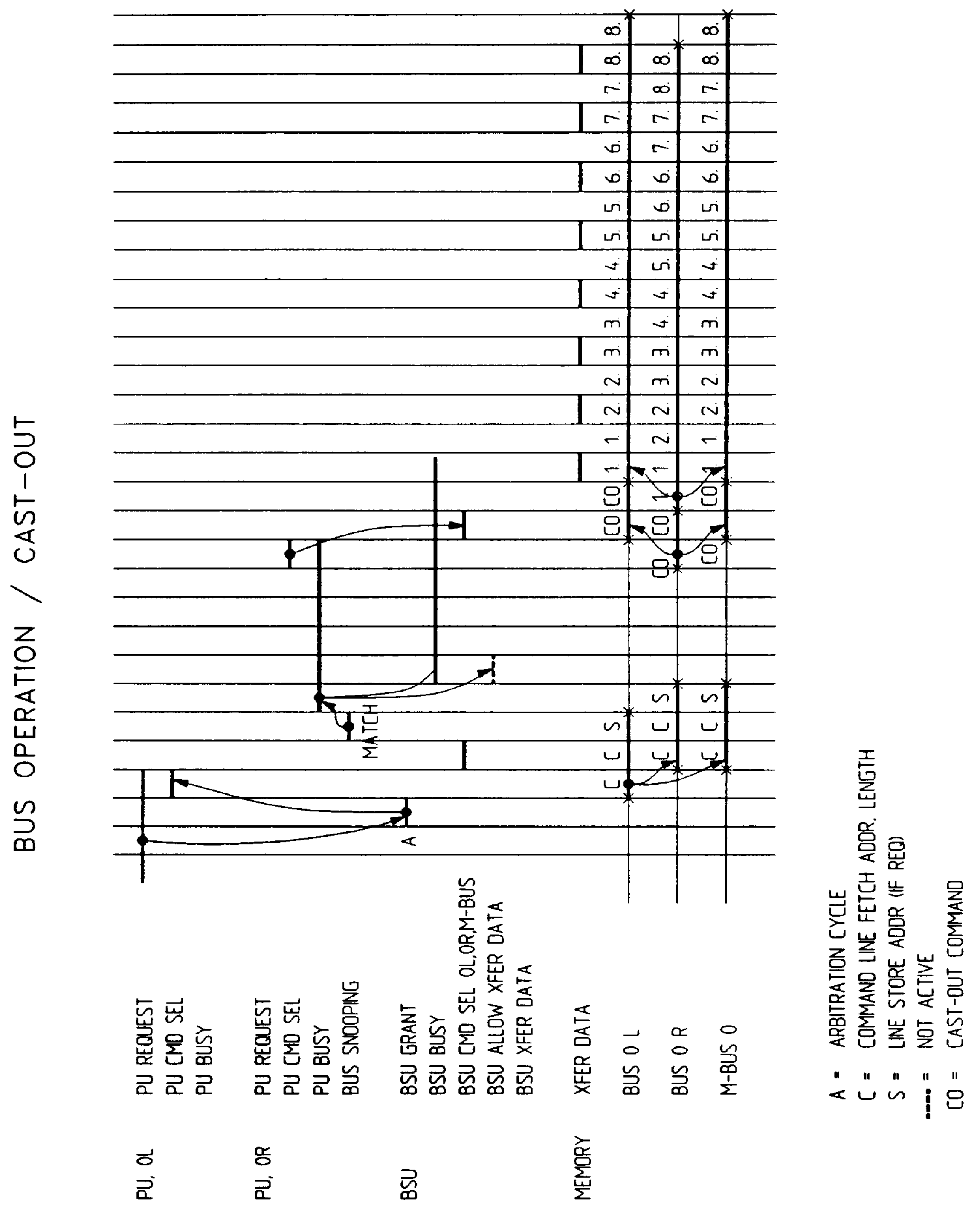

BUS OPERATION / CAST-OUT
PU, OL
PU REQUEST
PU CMD SEL
PU BUSY
PU, OR
PU REQUEST
PU CMD SEL
PU BUSY
BUS SNOOPING
BSU
BSU GRANT
BSU BUSY
BSU CMD SEL OL,OR,M-BUS
BSU ALLOW XFER DATA
BSU XFER DATA
MEMORY
XFER DATA
BUS 0 L
BUS 0 R
M-BUS 0
A
MATCH
C C S
C C S
C C S
CO CO 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8
CO CO 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8
CO CO 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8
A = ARBITRATION CYCLE
C = COMMAND LINE FETCH ADDR, LENGTH
S = LINE STORE ADDR (IF REQ)
---- = NOT ACTIVE
CO = CAST-OUT COMMAND

FIG. 9

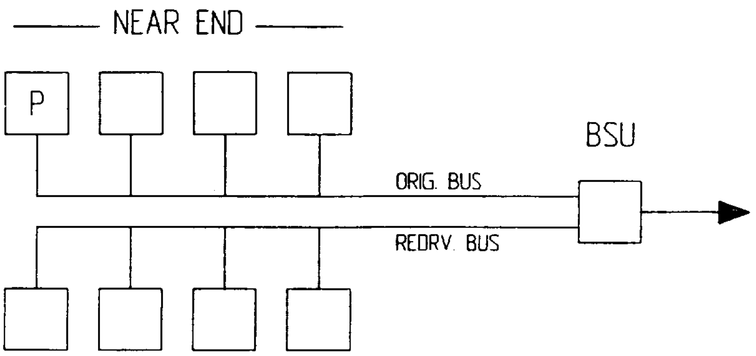

NEAR END
P
BSU
ORIG. BUS
REDRV. BUS

FIG. 10

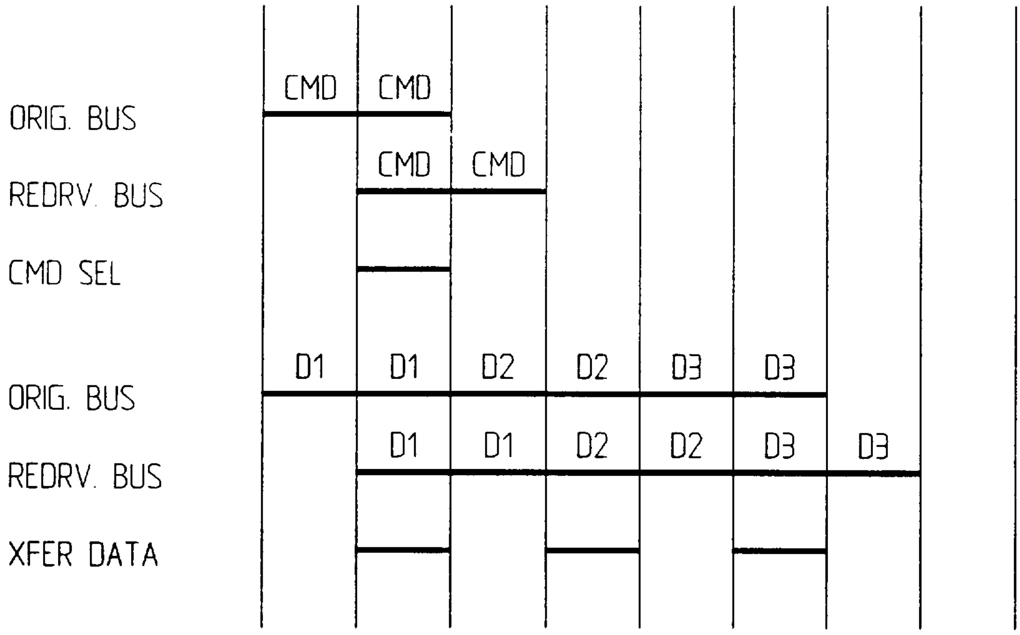

CMD
CMD
ORIG. BUS
CMD
CMD
REDRV. BUS
CMD SEL
D1
D1
D2
D2
D3
D3
ORIG. BUS
D1
D1
D2
D2
D3
D3
REDRV. BUS
XFER DATA

FIG. 11

INTERNATIONAL SEARCH REPORT

Inten. al Application No
PCT/EP 95/01140

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 G06F13/40 G06F13/16

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US,A,4 807 184 (SHELOR CHARLES F) 21 February 1989	1-5,8-11
A	see column 1, line 62 - column 2, line 18 see column 2, line 57 - column 4, line 58 see abstract; claims 1-3; figures 1,4 ---	6,12
X	EP,A,0 141 332 (SIEMENS AG) 15 May 1985	1-3,8,9
A	see page 2, line 23 - page 3, line 28 see page 4, line 34 - page 6, line 12 see page 6, line 25 - page 7, line 11 see abstract; claims 1-3; figures 1,4 ---	4-7, 10-12
A	US,A,5 337 411 (HARRISON JR EARNEST R) 9 August 1994 see column 2, line 55 - column 3, line 44 see abstract; figure 1 -----	1-12

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"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)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"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

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"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.

"&" document member of the same patent family

1

Date of the actual completion of the international search
7 December 1995

Date of mailing of the international search report
29.12.95

Name and mailing address of the ISA
European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+ 31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+ 31-70) 340-3016

Authorized officer
Nguyen Xuan Hiep, C

INTERNATIONAL SEARCH REPORT

Information on patent family members

Intern al Application No

PCT/EP 95/01140

Patent document cited in search report	Publication date	Patent family member(s)		Publication date
US-A-4807184	21-02-89	NONE		
EP-A-0141332	15-05-85	DE-A-	3338341	09-05-85
US-A-5337411	09-08-94	NONE		