PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 9/445, 13/00	A1	(11) International Publication Number: WO 95/22106
		(43) International Publication Date: 17 August 1995 (17.08.95)

(21) International Application Number: PCT/US95/01951

(22) International Filing Date: 9 February 1995 (09.02.95)

(30) Priority Data:
08/194,242 10 February 1994 (10.02.94) US

(71) Applicant: OAKLEIGH SYSTEMS, INC. [US/US]; Suite 110, 430 North Mary Avenue, Sunnyvale, CA 94086 (US).

(72) Inventor: KIKINIS, Dan; 20264 Ljepava Drive, Saratoga, CA 95070 (US).

(74) Agent: BOYS, Donald, R.; P.O. Box 187, Aromas, CA 95004 (US).

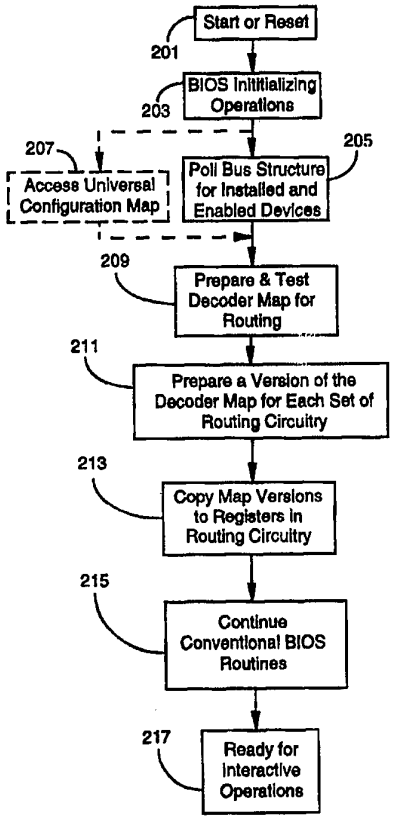
(81) Designated States: CN, JP, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.

(54) Title: I/O DECODER MAP

(57) Abstract

An I/O control system (19) for a general-purpose computer (13) having multiple bus branches (21) separated by routing circuitry (43, 44) prepares a digital map (48, 53) of I/O device locations on the bus branches, and copies versions of the maps to registers in the routing circuitry (213) on startup and reset (211). The routing maps provide immediate routing information for I/O requests issued by one or more CPUs (40, 41, 42) in a system, allowing therouting circuitry between bus branches to immediately route requests to the proper device with a minimum of wait states. In one aspect I/O devices are polled (205) for location at startup and reset (201), and in another aspect, a universal map protocol (207) is a part of the BIOS or storage accessible by the BIOS, making a system self-configuring and providing the necessary information for the mapping for the routing circuitry.



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

I/O DECODER MAP

Field of the Invention

This invention is in the field of methods and apparatus for bus management in computer systems.

Background of the Invention

An ability to communicate rapidly through bus architecture is a critical factor in personal computer performance. Early buses designed for IBM PC and XT computers were coupled directly to the CPU and operated in time with the CPU clock. Development of faster CPUs, however, such as with the inception of the IBM PC AT computer, increased CPU clock speed beyond the capabilities of most expansion devices. This led AT developers to design a system wherein the CPU and the bus ran with separate clocks at different speeds.

AT bus design proved so satisfactory that its structure became Industry Standard Architecture (ISA). Such a dual-speed configuration is common today, as most computers operate with slower buses and comply with conventional standards such as Extended Industry Standard Architecture (EISA) and Micro Channel Architecture (MCA).

In a quest for better performance from expansion devices, manufacturers have more recently refocused on bus architecture. Current state-of-the-art systems typically use a high-speed local bus for devices that have high data transfer rate capability, such as video adapters, hard drives, memory controllers, and network connections; and a slower expansion bus for devices that do not require or cannot operate at the higher rates.

In theory, a 32-bit local bus should be capable of transferring data as much as six times faster than an EISA or ISA bus, but most attempts at high-speed local bus implementation have had less than expected results. The first enhanced local bus products, dedicated solely to video systems, demonstrated only about 30 percent faster performance than an EISA or MCA expansion bus. Although part of the problem was found to be specific to video power and time requirements, the need for improvement was apparent. Since then several local bus standards have been developed and local buses are currently operable at clock cycle rates at or above 33 MHz.

A problem that still impedes optimum performance in bus management is the number of wait states that occur before a CPU can locate a peripheral device on the bus. Because a CPU typically has no way of knowing the port assignment of a peripheral device, the CPU must transmit an address request to find the device on the bus. In many cases there will be more than one branch of a local bus, because the ability of a conventional CPU to simultaneously drive several loads is limited. Branches are separated by chip sets, which comprise solid state circuitry for buffering and routing addresses and data from branch to branch. Each time buffering occurs, wait states have to be added and operation is slowed.

If, after a request is made, the addressed device does not respond within a certain period of time, the CPU goes on to try to find the device at another address. Each time a device is not at an address, the CPU adds wait states.

What is needed is a means to decrease wait states by giving a CPU immediate access to device addresses on a bus architecture. Such a bus system might also be self-configuring so a user can plug a device into any slot and immediately use it.

Summary of the Invention

In an embodiment of the invention, a method for routing requests for I/O devices is provided for use with a computer having a bus structure with branches separated by sets of routing circuitry (typically chip sets for branching bus architecture). The method comprises steps of (a) preparing a digital map on startup and reset for portraying the location of the devices on the branches of the bus structure; (b) copying versions of the digital map to registers in the sets of routing circuitry, each map version being particular to the set of routing circuitry for which it is prepared; and (c) routing requests for I/O devices at each set of routing circuitry according to the digital map copied to that set of routing circuitry.

In one aspect of the invention, the information needed for composing the digital maps is obtained by polling the bus structure I/O devices on startup and reset to ascertain the location of installed devices on the bus branches. In another aspect each I/O port available in a system configuration is assigned a universal configuration port number, providing a universal map protocol stored in a nonvolatile memory device, and the universal map protocol is read by the system BIOS on startup and reset to prepare the digital map.

Typical maps according to embodiments of the invention for copying to chip sets comprise a row assigned to each I/O device, a bit column indicating I/O devices installed and enabled, and one or more bit columns indicating the branch connected to the routing circuitry associated with the map for routing received I/O requests. For chip sets having two bus ports, the map typically has just two bit columns.

Polling, map preparation, and the like are controlled in one embodiment by control routines that are a part of the system BIOS.

In an alternative embodiment the control routines may be a part of a separate non-volatile storage accessible by the BIOS on start up and reset. In yet another embodiment an interactive display may be provided for a user to enter port assignments on bus branches for installed I/O devices, the port assignments then being stored and used for preparing the digital map.

Brief Description of the Drawings

Fig. 1A is a block diagram of a local bus structure in combination with an I/O expansion bus according to an embodiment of the present invention.

Fig. 1B is an illustration of a bus decoder map according to the embodiment of Fig. 1A.

Fig. 2A is a block diagram of a bus structure shared by multiple CPUs in a variation of the embodiment shown in Fig. 1A.

Fig. 2B is an illustration of a bus decoder map for one chip set according to the embodiment of Fig. 2A.

Fig. 2C is an illustration of a decoder map for the second chip set of Fig. 2A.

Fig. 3A is a block diagram of a multiple-CPU bus structure in another aspect of the invention.

Fig. 3B shows a decoder map for one of the chip set ports according to the embodiment of Fig. 3A.

Fig. 3C shows a decoder map for another of the chip set ports according to the embodiment of Fig. 3A.

Fig. 3D shows a decoder map for another of the chip set ports according to the embodiment of Fig. 3A.

Fig. 3E shows a decoder map for another of the chip set ports according to the embodiment of Fig. 3A.

Fig. 3F shows a decoder map for another of the chip set ports according to the embodiment of Fig. 3A.

Fig. 3G shows a decoder map for another of the chip set ports according to the embodiment of Fig. 3A.

Fig. 3H shows a decoder map for another of the chip set ports according to the embodiment of Fig. 3A.

Fig. 3I shows a decoder map for another of the chip set ports according to the embodiment of Fig. 3A.

Fig. 4A is a block diagram for a bus structure having multiple CPUs connected to a single chip set.

Fig. 4B is an illustration of a decoder map for one port of the chip set of Fig. 4A.

Fig. 5A is a block diagram of a bus structure having a single CPU connected to a single chip set with multiple bus branches.

Fig. 5B is an illustration of a decoder map for the embodiment of Fig. 5A.

Fig. 6 is a logic flow diagram illustrating steps in developing and using decoder maps according to an embodiment of the invention.

Description of the Preferred Embodiments

Fig. 1A is a block diagram for a bus architecture comprising a local bus structure 13 in combination with an I/O expansion bus 21 according to an embodiment of the present invention. CPU 11 is connected through local bus 13 to chip set 19 and to local devices 15, 17, and 18. Chip set 19 is connected through I/O expansion bus 21 to I/O devices 23 and 25.

In this embodiment of the invention, a control routine, which may reside in BIOS or may be accessed by BIOS from another

source, polls all devices at startup and reset and sets up a decoder map in system RAM (not shown) or other memory area for testing. Once tested, the decoder map is copied into registers (not shown) in chip set 19. When CPU 11 initiates communication with a device, chip set 19 immediately identifies address and routing information from the decoder map and routes the request in one wait state. There are no extra wait states as there would be if the CPU had to search for the device address.

Fig. 1B is a diagram of a bus decoder map 33 for chip set 19 of Fig. 1A. Decoder map 33 has a maximum address range 34 in this embodiment of 1-256. There could be more or fewer addresses in the range in other embodiments. Addresses 1 through 3 store routing information for local devices 15, 17, and 18, respectively (Fig. 1A), addresses 4 and 5 store routing information for devices 23 and 25. Column 35 stores a bit denoting an enabled device. The bit in column 35 is arbitrarily set as 1 if a port is enabled, or to 0 if a port is not enabled. The designation could as well be opposite. Second bit 37 is arbitrarily set to 1 if the device referenced is on one side of chip set 19 (on the local bus, for example), or 0 if the device is on the other side (in this case on expansion bus 21). This convention is arbitrary and could be reversed, as well.

Because in this example only a single local bus with no branching exists, all second bits 37 for addresses 1 through 3 are set the same. Devices 23 and 25 on the expansion bus are represented in the map of Fig. 1B as number 4 and 5, where the enable bit is 1 and the routing bit is also 1, indicating these devices are on the opposite side of chip set 19 from devices 15, 17, and 18. The convention here, assuming a chip set with two bus ports, one to the local bus and one to the expansion bus, is to indicate with the routing bit that the device is either on the local side or on the expansion side.

In the embodiment of Fig. 1A, at startup or reset, as indicated above, devices are polled, a routing map is prepared and tested, and the map is copied to available registers of chip set 19. Thereafter, the chip set acts as a bridge, switching appropriately when necessary to route requests generated by CPU 11 to far side expansion bus 21 or to near side local bus 13.

The performance improvement in the case of the architecture of Fig. 1A is real, but somewhat marginal, because of the inherent simplicity of the architecture. In more sophisticated bus architectures, particularly in architectures having branched, high-speed, local buses, the performance improvement is much greater.

Fig. 2A is a block diagram of a branched bus configuration, wherein three CPUs 40, 41, and 42 share primary bus branch 45 and two secondary bus branches 46 and 47. In this configuration CPU's 40, 41, and 42 are connected to two chip sets 43 and 44. In this embodiment, on startup and reset, decoder maps developed by polling are loaded to each of chip sets 43 and 44, providing complete routing information for all bus branches.

In this example, primary bus branch 45 has 8 device slots with addresses 1 to 8; secondary bus branches 46 and 47 have four device slots each with addresses 9 to 12 and 13 to 16, respectively. Slots having enabled devices installed are circled on the diagram.

In this embodiment, if CPU 42 addresses device 10 on bus branch 46, a request is quickly routed to device 10 through chip set 44 by virtue of the decoder maps being stored in both chip sets 43 and 44. Only one wait state is expended instead of the three wait states or more it would take for a conventional bus configuration to go to both chip sets to search the different bus branches for the device.

Figs. 2B and 2C are illustrations of decoder maps 48 and 53 for chip sets 43 and 44, respectively, of Fig. 2A. Both of these

decoder maps have maximum address ranges 49 and 54 of 1-256. The bit in columns 50 and 55 is the enable bit and the bit in columns 51 and 56 is the routing bit. Criteria for setting these two bits is the same as that described for the decoder map in Fig. 1B. For example, map 48 for chip set 43 has a 1 in enable column 50 for all the circled devices, and a 1 in routing column 51 for devices 9, 10, and 11, which are on the expansion side of chip set 43.

Fig. 3A is a block diagram 57 of another variation of a bus configuration, wherein four CPUs 59, 61, 63, and 65 are associated with four chip sets 67, 69, 71, and 73. There are five bus segments 75, 77, 79, 81, and 82, which communicate with the chip sets at ports 58, 60, 62, 64, 66, 68, 70, and 72. Each of bus branches 75, 77, 79, and 81 have four device positions. Positions with enabled devices installed are shown circled. In this embodiment the routing maps for each chip set will still be two-dimensional, having one bit for enable and another for routing. There will be, however, an additional complication.

Consider a request by CPU 59 for device 14. This request intercepted at port 58 needs a map that directs all requests for other than devices 1, 2, and 3 to bus branch 82. This is not sufficient, however, because once the request is on branch 82, it needs to be directed at ports 64, 68, and 72. So every port on the chip sets (in this case) needs a map. There will therefore be two maps in each chip set: one for each bus port.

Figs. 3B, 3C, 3D, 3E, 3F, 3G, 3H, and 3I are two-dimensional decoder maps for ports 58, 60, 62, 64, 66, 68, 70, and 72, respectively. For the eight two-dimensional maps in this embodiment, the bits in the enable column for the installed and enabled devices (circled in Fig. 3A) will be the same. The bit for the route column for each two-dimensional map will reflect whether the device sought

is on the port side or the opposite side from the port associated with the particular map.

There are alternatives to the two-dimensional maps, port by port, described above for the embodiment of Fig. 3A. There could be a single map for each chip set, for example, with a routing column for each branch having installed devices. In this case, the routing circuitry at each chip set would be configured to switch appropriately in response to a 1 bit in a routing column for an installed device.

Fig. 4A illustrates a system 121 having four branches of a bus structure, four CPUs, and one chip set. CPU 123 connects to chip set 139 through branch 131, CPU 125 connects to chip set 139 through branch 133, CPU 127 connects to chip set 139 through branch 135, and CPU 129 connects to chip set 139 through branch 137. There are three enabled devices (1), (2), and (3) on branch 131, three enabled devices (5), (6), and (7) on branch 133, three enabled devices (9), (10), and (11) on branch 135, and three enabled devices (13), (14), and (15) on branch 137.

In the embodiment of Fig. 4A, the chip set has four ports, labeled (A), (B), (C), and (D). The chip set, then, assuming all the bus branches have one or more enabled devices installed, must route requests from any one CPU to three alternative destinations.

In this embodiment, there are four maps, one for each port. Each map has an enable column and three routing columns. Fig. 4B shows the map for port (A) of Fig. 4A. Devices 1-3, 5-7, 9-11, and 13-15 are enabled, so the bit for enable column 141 is 1 for these devices. This will be common for the four maps.

For port (A), devices 1, 2, and 3 are CPU direct, so routing columns 143, 145, and 147 are all zero for these devices. Arbitrarily, a 1 bit is assigned to one of the three routing columns to indicate each of the three "other" branches for port (A). In this case column 143 is

assigned for branch 133, column 145 for branch 135, and column 147 for branch 137. A 1 bit in a column indicates the device in that row is enabled on that bus branch. So for device 10, for example, in the map for port (A), column 145 has a 1 bit and columns 143 and 147 are zero.

Given the example of Fig. 4B, the maps for the other ports are intuitive and are not reproduced here.

There are a number of other ways mapping may be accomplished for the example of Fig. 4A. There could be, for example, one map, and logic could be provided so the entry port for a request for a device would rearrange the map to suit the port. In general, there needs to be a map, or a perturbation of the map, for each CPU port, and the routing columns need to combine the bits in a unique fashion to indicate the branch for routing depending on the enabled device. Having a unique bit for each "other" port is one way. A map could also have a unique routing column for each of the four ports. Another solution is to take advantage of the minimum number of unique bit patterns to indicate routing. In this solution two columns may indicate four branches. 00 is CPU side, 01 is one alternative branch, 10 is another, and 11 is the third alternative branch, for four total. Three bits may then indicate eight branches, and so on.

Fig. 5A is a block diagram of a system 173 having a single CPU 175, a single chip set 179, and four bus branches 177, 181, 183, and 185, one of which (177) is the CPU branch. Fig. 5B shows a routing map for this case, having three routing columns 191, 193, and 195. As described above for the map for the embodiment of Fig. 4A, all routing columns 0 means the device requested is on the CPU branch. A 1 bit in column 191 routes requests by CPU 175 to branch 181; a 1 bit in column 193 routes requests to branch 183, and a 1 bit

in column 195 routes requests to branch 185.

As described above for the embodiment of Fig. 4A, there are several alternatives for the map shown for the embodiment of Fig. 5A. There could be four routing columns, for example, one for each branch, or the columns could be used as bit patterns to indicate routing.

In an alternative embodiment, a system having a bus structure such as that described above for Fig. 1A, Fig 2A, Fig. 3A, Fig. 4A, Fig. 5A, or other branching structure not shown, may be self-configuring. That is, instead of BIOS having to poll devices at setup, each type of device on the bus is preassigned a universal configuration port number through a programmable I/O map protocol. This protocol can be stored in a Complementary Metal Oxide Semiconductor (CMOS), Erasable Programmable Read-Only Memory (EPROM), or other suitable type of memory chip, and is accessed by BIOS on startup to initialize the bus decoder map. With this innovation a user can install a device in any location on a branched bus structure and use the device immediately without need to set up a port assignment.

In another embodiment, a bus structure such as that described for Fig. 1A, Fig. 2A, Fig. 3A, Fig. 4A, Fig. 5A, or other variation, can be configured by a user through a control routine that provides a means to select device assignments from an on-screen menu interface, and the assignments will be copied to a permanent memory area that is accessed by BIOS at startup.

Fig. 6 is a logic flow diagram showing steps involved in an embodiment of the present invention for initializing a decoder map and providing the map to routing circuitry in a branched bus structure. At step 201 a signal is received for startup or reset of a computer system configured according to the invention. Initially, conventional BIOS routines execute, for such as memory initialization and the like,

and these activities are represented as step(s) 203.

At some point, routines execute in the BIOS (or, in an alternative case, are accessed by the BIOS and execute), for the map initialization as described in several embodiments above. In one embodiment the bus structure is polled to locate installed and enabled devices, and the devices are assigned a number and location for a map. This polling step is indicated in Fig. 6 as step 205. Alternatively, the BIOS may access a universal map protocol for the information, represented by alternative step 207 shown in dotted outline with dotted flow path.

Once the necessary information is obtained, a map is prepared and tested at step 209. In this step, a copy of the map may be stored in system RAM, or other system memory. At step 211 versions of the decoder map are prepared for the routing circuitry separating bus branches in the system. At step 213, the map versions are copied to the appropriate routing circuitry. This circuitry is typically implemented in solid state in one or more chips, and is typically referred to as a chip set.

Step 215 indicates continuation of BIOS functions after the map preparation, according to embodiments of the invention, is complete. At some point, indicated by step 217, initialization is complete, and the system is ready to begin interactive operations, typically involving loading a specific application, such as a word processing application. During operation, requests for bus devices will be routed by various chip sets separating buses and bus branches according to the decoder maps prepared and copied to the chip sets.

It will be apparent to one with skill in the art that there are a relatively large number of changes that may be made in the embodiments described above without departing from the spirit and scope of the invention. Other variations might enhance bus

performance by incorporating other currently available optimization features into the architecture. Many other alternatives can fall within the scope and spirit of the invention.

What is claimed is:

1. In a computer having a bus structure with branches separated by sets of routing circuitry, a method for routing requests for I/O devices, comprising steps of:

(a) preparing a digital map on startup and reset for portraying the location of the devices on the branches of the bus structure;

(b) copying versions of the digital map to registers in the sets of routing circuitry, each map version being particular to the set of routing circuitry for which it is prepared; and

(c) routing requests for I/O devices at each set of routing circuitry according to the digital map copied to that set of routing circuitry.

2. The method of claim 1 comprising steps for polling I/O devices on startup and reset to ascertain the location of installed devices on the bus branches, and using the poll results in the preparing step.

3. The method of claim 1 wherein each I/O port available in a system configuration is assigned a universal configuration port number, providing a universal map protocol stored in a nonvolatile memory device, and the universal map protocol is read by the system Basic Input Output System (BIOS) on startup and reset to prepare the digital map.

4. The method of claim 1 wherein the digital maps prepared comprise a row assigned to each I/O device, a bit column indicating I/O devices installed and enabled, and one or more bit columns indicating the branch connected to the routing circuitry associated with the map for routing received I/O requests.

5. The method of claim 2 wherein control routines that are a part of the Basic Input Output System control the polling, map preparation, and copying of maps to sets of routing circuitry.

6. The method of claim 2 wherein control routines not a part of the Basic Input Output System are accessed by the Basic Input Output System on startup and reset for controlling the polling, map preparation, and copying of maps to sets of routing circuitry.

7. The method of claim 1 wherein a set of routing circuitry has two bus ports, and the digital map copied to that set has one routing column wherein a one bit in the routing column in a row indicates the addressed device assigned to that row is on one side of the set of routing circuitry, and a zero bit at the same position means the addressed device is on the other side of the set.

8. The method of claim 1 wherein a set of routing circuitry has more than two bus ports connecting more than two bus branches, and the digital map copied to that set has plural routing columns.

9. The method of claim 1 wherein an interactive, on-screen interface is provided on a display coupled to the computer, the interactive interface providing for user entry of port assignments on bus branches for installed I/O devices, the port assignments stored and used in the step for preparing the digital map.

10. A Basic Input Output System (BIOS) for a general-purpose computer having a branched bus structure with branches separated by sets of routing circuitry, comprising:

a non-volatile digital storage device connectable to a computer

bus;

initialization code sequences for initializing computer elements including volatile memory on startup and reset of the general-purpose computer;

input/output code sequences for controlling communication with devices connected to the general-purpose computer; and

mapping code sequences for preparing a digital map on startup and reset of the general-purpose computer for portraying the location of input/output devices installed on branches of bus structures separated by the sets of routing circuitry, and for copying versions of the digital map to the sets of routing circuitry.

11. A Basic Input Output System as in claim 10 wherein the mapping code sequences are configured for polling all bus ports on the bus branches for ascertaining the location of installed devices, and for using the results of the polling operation to prepare the digital map.

12. A Basic Input Output System as in claim 10 wherein each I/O port available in a system configuration is assigned a universal configuration port number, providing a universal map protocol stored in the Basic Input Output System, and the universal map protocol is used for preparing the digital map.

13. A Basic Input Output System as in claim 12 wherein the universal map protocol is stored in a programmable read-only memory separate from the Basic Input Output System, which memory is accessed by the Basic Input Output System on startup and reset to prepare the digital map.

14. A Basic Input Output System as in claim 10 wherein the versions

of the digital map prepared comprise a row assigned to each I/O device, a bit column indicating I/O devices installed and enabled, and one or more bit columns indicating the branch connected to the routing circuitry associated with the map for routing received I/O requests.

15. A Basic Input Output System as in claim 10 wherein a set of routing circuitry has two bus ports, and the digital map version copied to that set has one routing column wherein a one bit in the routing column in a row indicates the addressed device assigned to that row is on one side of the set of routing circuitry, and a zero bit at the same position means the addressed device is on the other side of the set.

16. A Basic Input Output System as in claim 10 wherein a set of routing circuitry has more than two bus ports connecting more than two bus branches, and the digital map copied to that set has plural routing columns.

17. A general-purpose computer system comprising:

a CPU;

a bus structure coupled to the CPU wherein bus branches having connections for I/O devices are separated by and communicate through sets of routing circuitry; and

a Basic Input Output System coupled to the CPU for providing basic operating instructions for the CPU;

wherein the Basic Input Output System is configured for preparing a digital map on startup and reset portraying the location of input/output devices installed on the bus branches, and for copying versions of the digital map to the sets of routing circuitry.

18. A general-purpose computer system as in claim 17 wherein the Basic Input Output System is configured for polling all bus ports on the bus branches for ascertaining the location of installed devices, and for using the results of the polling operation to prepare the digital map.

19. A general-purpose computer system as in claim 17 wherein each I/O port available in a system configuration is assigned a universal configuration port number, providing a universal map protocol stored in the Basic Input Output System, and the universal map protocol is used for preparing the digital map.

20. A general-purpose computer as in claim 19 wherein the universal map protocol is stored in a programmable read-only memory separate from the Basic Input Output System, which memory is accessed by the Basic Input Output System on startup and reset to prepare the digital map.

21. A general-purpose computer as in claim 17 wherein the versions of the digital map prepared comprise a row assigned to each I/O device, a bit column indicating I/O devices installed and enabled, and one or more bit columns indicating the branch connected to the routing circuitry associated with the map for routing received I/O requests.

22. A general-purpose computer system as in claim 21 wherein a set of routing circuitry has two bus ports, and the digital map version copied to that set has one routing column wherein a one bit in the routing column in a row indicates the addressed device assigned to that row is on one side of the set of routing circuitry, and a zero bit at

the same position means the addressed device is on the other side of the set.

23. A general-purpose computer as in claim 21 wherein a set of routing circuitry has more than two bus ports connecting more than two bus branches, and the digital map copied to that set has plural routing columns.

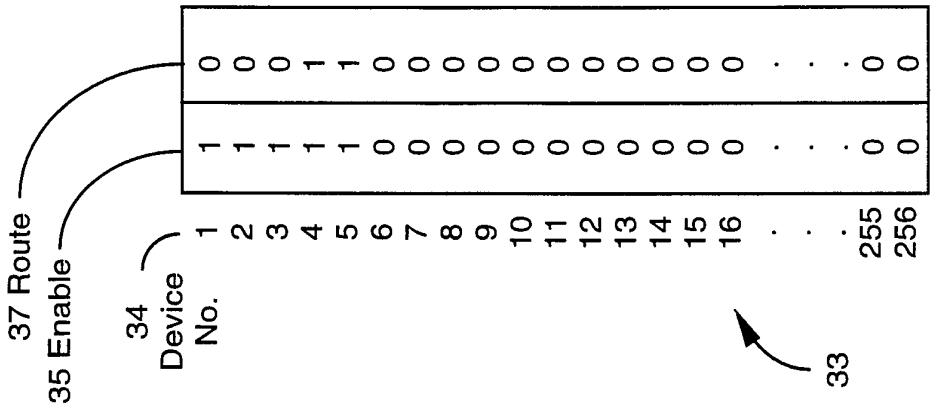

37 Route
35 Enable
34 Device No.
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 . . . 255 256
1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 . . . 0 0
0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 . . . 0 0
33

Fig. 1B

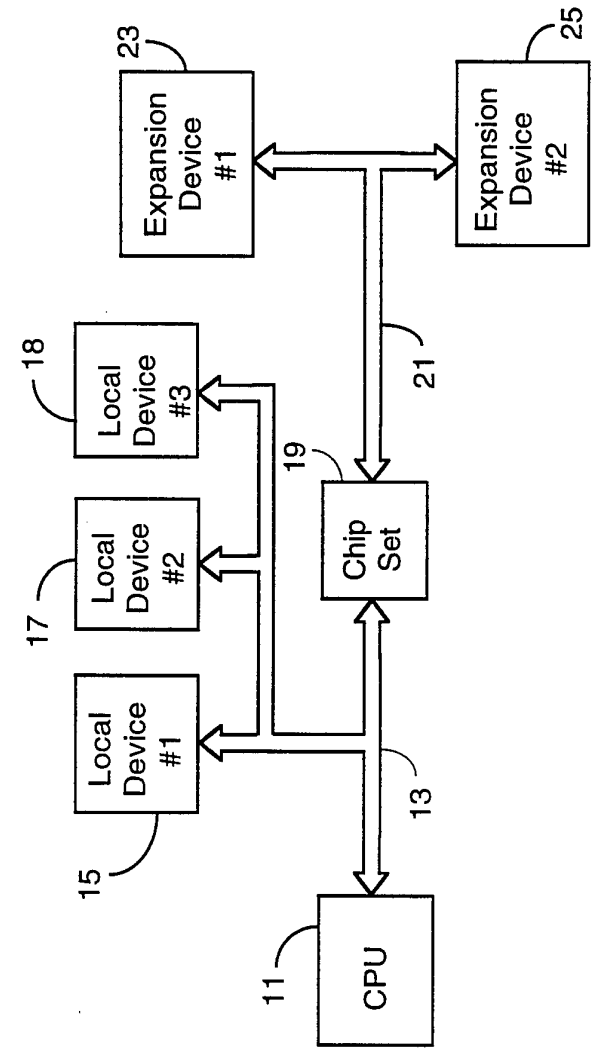

Expansion Device #1
23
Expansion Device #2
25
Local Device #3
18
Local Device #2
17
Local Device #1
15
Chip Set
19
CPU
11
13
21

Fig. 1A

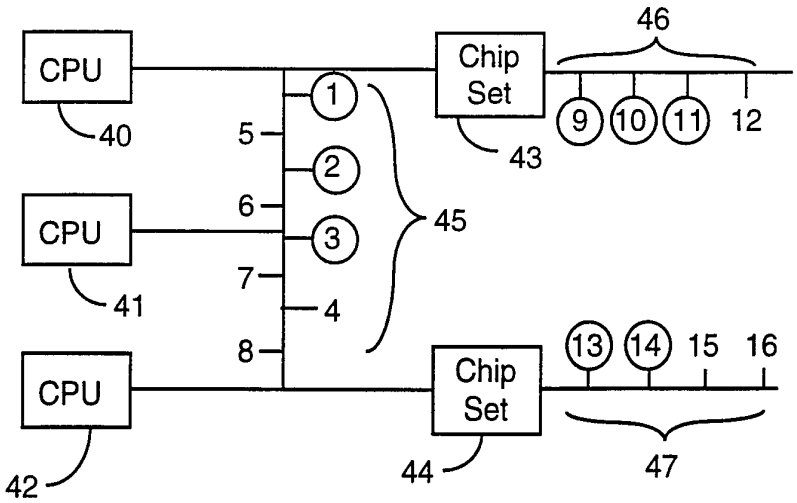
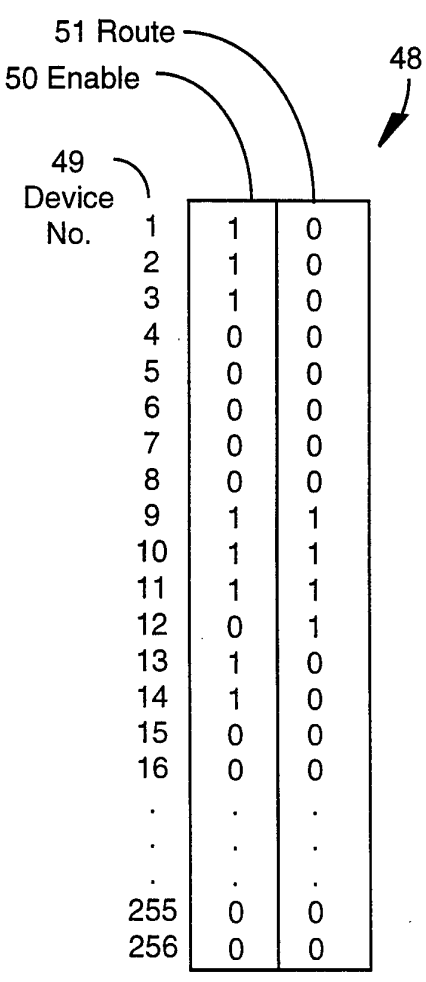
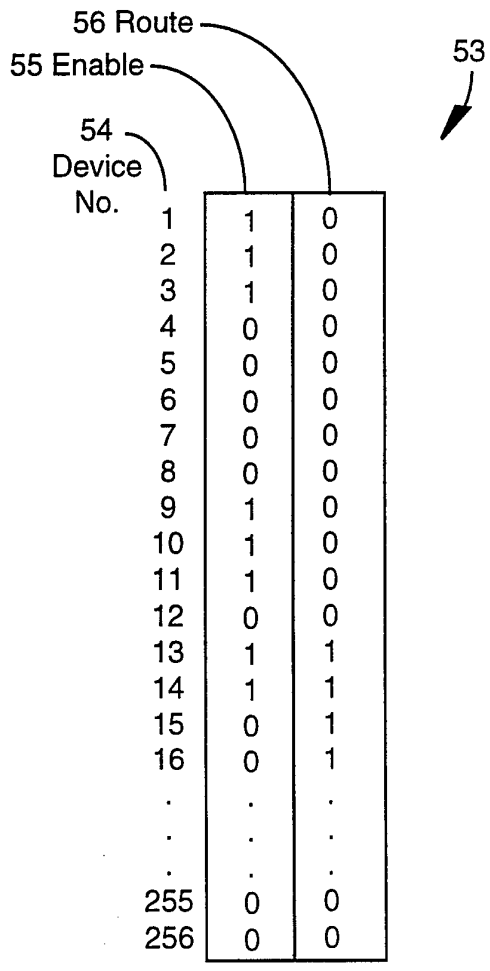


Fig. 2A



Device No.		
1	1	0
2	1	0
3	1	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	1	1
10	1	1
11	1	1
12	0	1
13	1	0
14	1	0
15	0	0
16	0	0
.	.	.
.	.	.
.	.	.
255	0	0
256	0	0

Fig. 2B



Device No.		
1	1	0
2	1	0
3	1	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	1	0
10	1	0
11	1	0
12	0	0
13	1	1
14	1	1
15	0	1
16	0	1
.	.	.
.	.	.
.	.	.
255	0	0
256	0	0

Fig. 2C

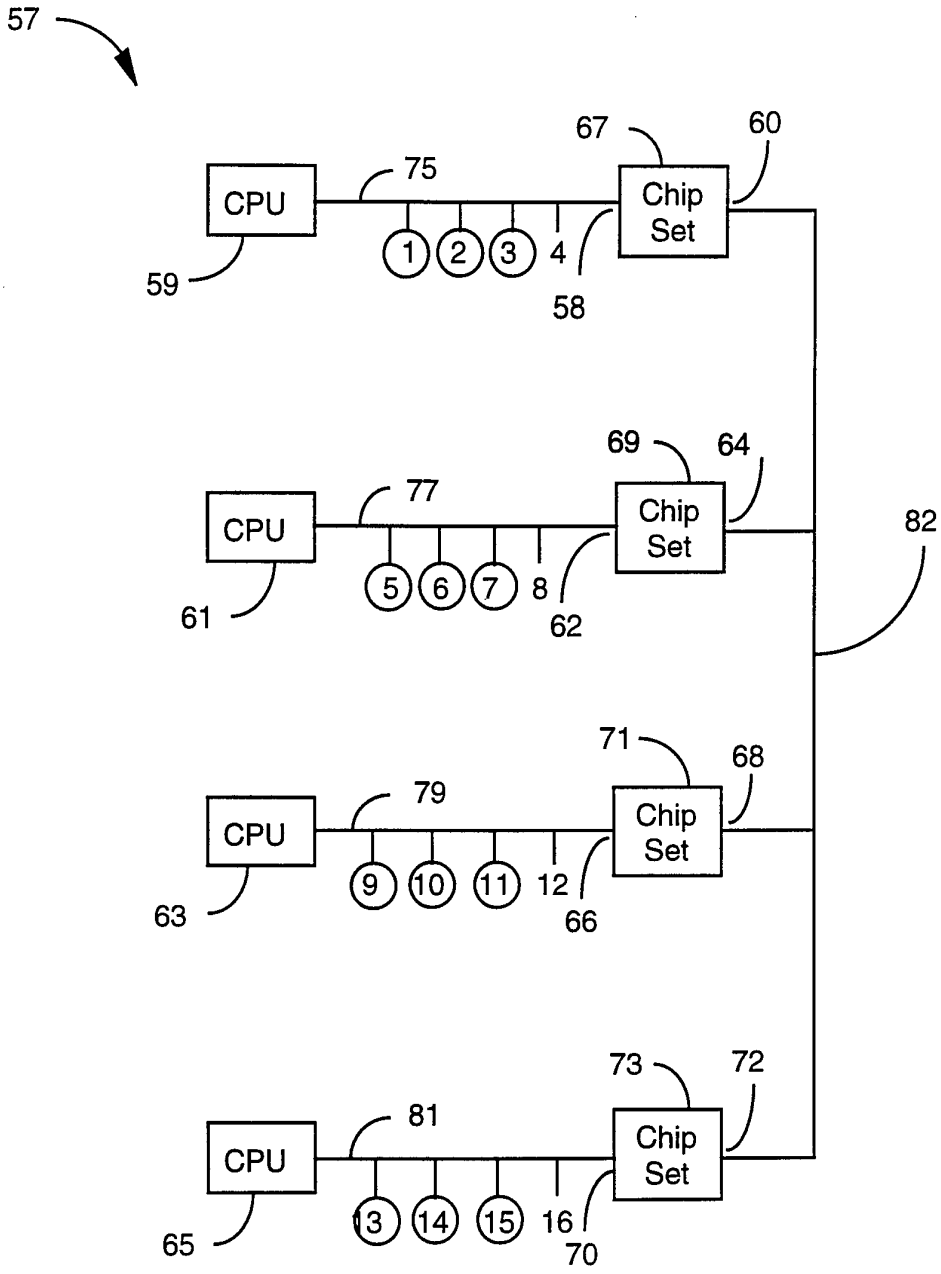

57
67
60
CPU
75
Chip Set
1
2
3
4
59
58
69
64
CPU
77
Chip Set
82
5
6
7
8
61
62
71
68
CPU
79
Chip Set
9
10
11
12
63
66
73
72
CPU
81
Chip Set
13
14
15
16
65
70

Fig. 3A

Device	Enable	Route
1	1	0
2	1	0
3	1	0
4	0	0
5	1	1
6	1	1
7	1	1
8	0	1
9	1	1
10	1	1
11	1	1
12	0	1
13	1	1
14	1	1
15	1	1
16	0	1
.	.	.
.	.	.
.	.	.
255	0	0
256	0	0

Fig. 3B

Device	Enable	Route
1	1	1
2	1	1
3	1	1
4	0	1
5	1	0
6	1	0
7	1	0
8	0	0
9	1	0
10	1	0
11	1	0
12	0	0
13	1	0
14	1	0
15	1	0
16	0	0
.	.	
.	.	
.	.	
255	0	
256	0	

Fig. 3C

Device	Enable	Route
1	1	1
2	1	1
3	1	1
4	0	1
5	1	0
6	1	0
7	1	0
8	0	0
9	1	1
10	1	1
11	1	1
12	0	1
13	1	1
14	1	1
15	1	1
16	0	1
.	.	.
.	.	.
.	.	.
255	0	0
256	0	0

Fig. 3D

Device	Enable	Route
1	1	0
2	1	0
3	1	0
4	0	0
5	1	1
6	1	1
7	1	1
8	0	1
9	1	0
10	1	0
11	1	0
12	0	0
13	1	0
14	1	0
15	1	0
16	0	0
.	.	
.	.	
.	.	
255	0	
256	0	

Fig. 3E

Device	Enable	Route
1	1	1
2	1	1
3	1	1
4	0	1
5	1	1
6	1	1
7	1	1
8	0	1
9	1	0
10	1	0
11	1	0
12	0	0
13	1	1
14	1	1
15	1	1
16	0	1
.	.	.
.	.	.
.	.	.
255	0	0
256	0	0

Fig. 3F

Device	Enable	Route
1	1	0
2	1	0
3	1	0
4	0	0
5	1	0
6	1	0
7	1	0
8	0	0
9	1	1
10	1	1
11	1	1
12	0	1
13	1	0
14	1	0
15	1	0
16	0	0
.	.	
.	.	
.	.	
255	0	
256	0	

Fig. 3G

Device	Enable	Route
1	1	1
2	1	1
3	1	1
4	0	1
5	1	1
6	1	1
7	1	1
8	0	1
9	1	1
10	1	1
11	1	1
12	0	1
13	1	0
14	1	0
15	1	0
16	0	0
.	.	
.	.	
.	.	
255	0	
256	0	

Fig. 3H

Device	Enable	Route
1	1	0
2	1	0
3	1	0
4	0	0
5	1	0
6	1	0
7	1	0
8	0	0
9	1	0
10	1	0
11	1	0
12	0	0
13	1	1
14	1	1
15	1	1
16	0	1
.	.	
.	.	
.	.	
255	0	
256	0	

Fig. 3I

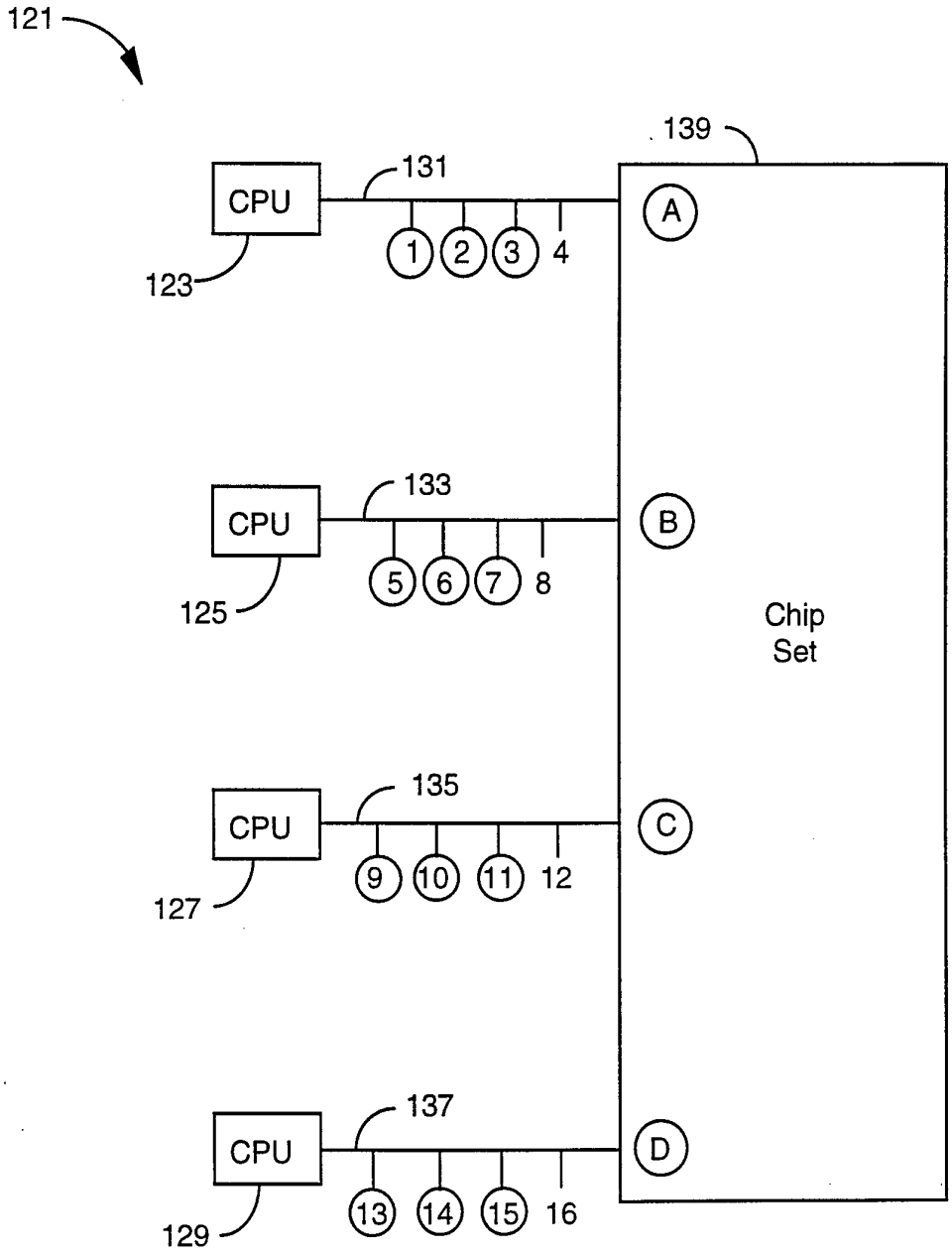
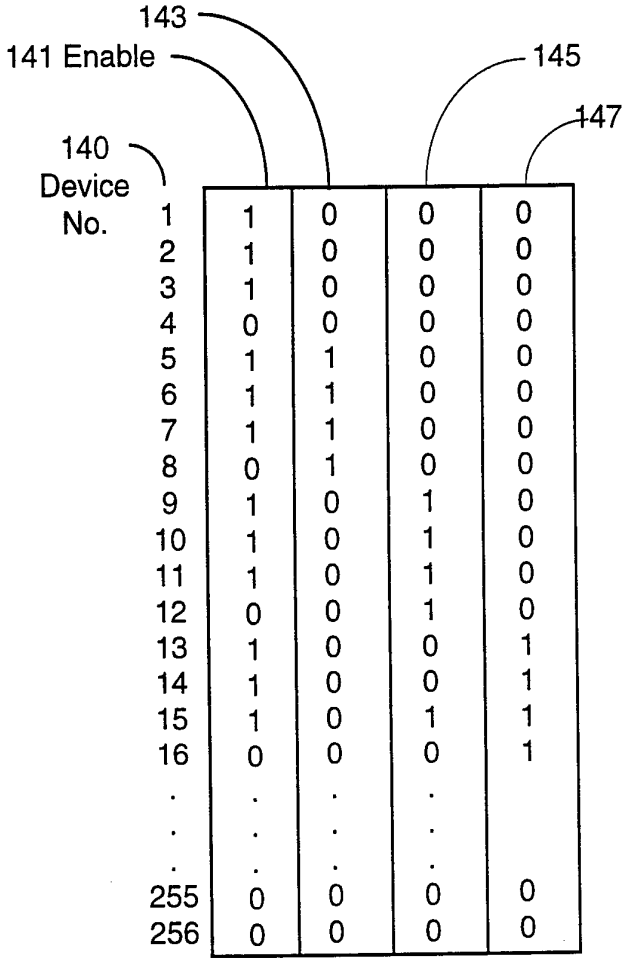

121
139
CPU
131
1
2
3
4
A
123
CPU
133
5
6
7
8
B
125
Chip
Set
CPU
135
9
10
11
12
C
127
CPU
137
13
14
15
16
D
129

Fig. 4A



Device No.	141 Enable	143	145	147
1	1	0	0	0
2	1	0	0	0
3	1	0	0	0
4	0	0	0	0
5	1	1	0	0
6	1	1	0	0
7	1	1	0	0
8	0	1	0	0
9	1	0	1	0
10	1	0	1	0
11	1	0	1	0
12	0	0	1	0
13	1	0	0	1
14	1	0	0	1
15	1	0	1	1
16	0	0	0	1
.	.	.	.	
.	.	.	.	
.	.	.	.	
255	0	0	0	0
256	0	0	0	0

Fig. 4B

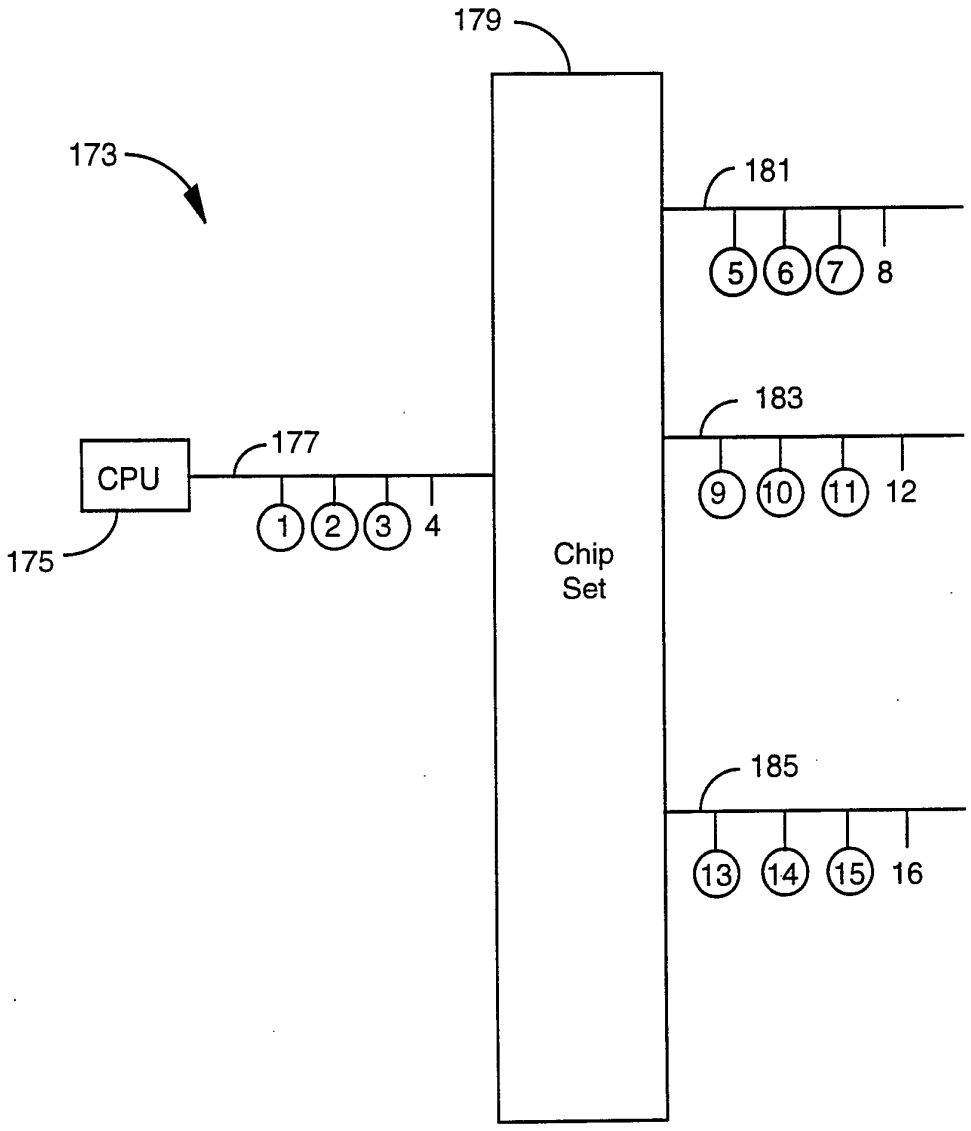
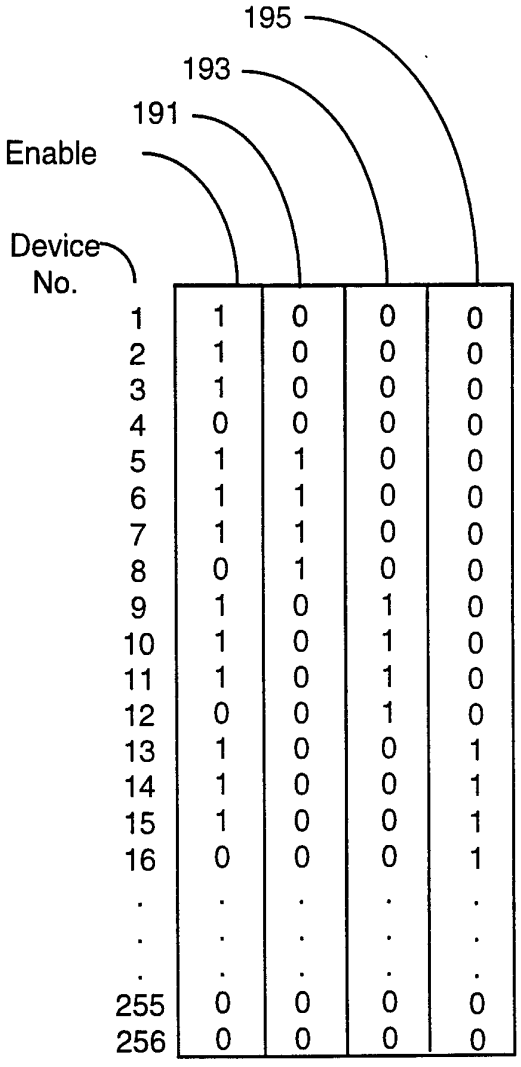

179
173
181
5
6
7
8
183
CPU
177
9
10
11
12
1
2
3
4
175
Chip
Set
185
13
14
15
16

Fig. 5A



Device No.	Enable	191	193	195
1	1	0	0	0
2	1	0	0	0
3	1	0	0	0
4	0	0	0	0
5	1	1	0	0
6	1	1	0	0
7	1	1	0	0
8	0	1	0	0
9	1	0	1	0
10	1	0	1	0
11	1	0	1	0
12	0	0	1	0
13	1	0	0	1
14	1	0	0	1
15	1	0	0	1
16	0	0	0	1
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
255	0	0	0	0
256	0	0	0	0

Fig. 5B

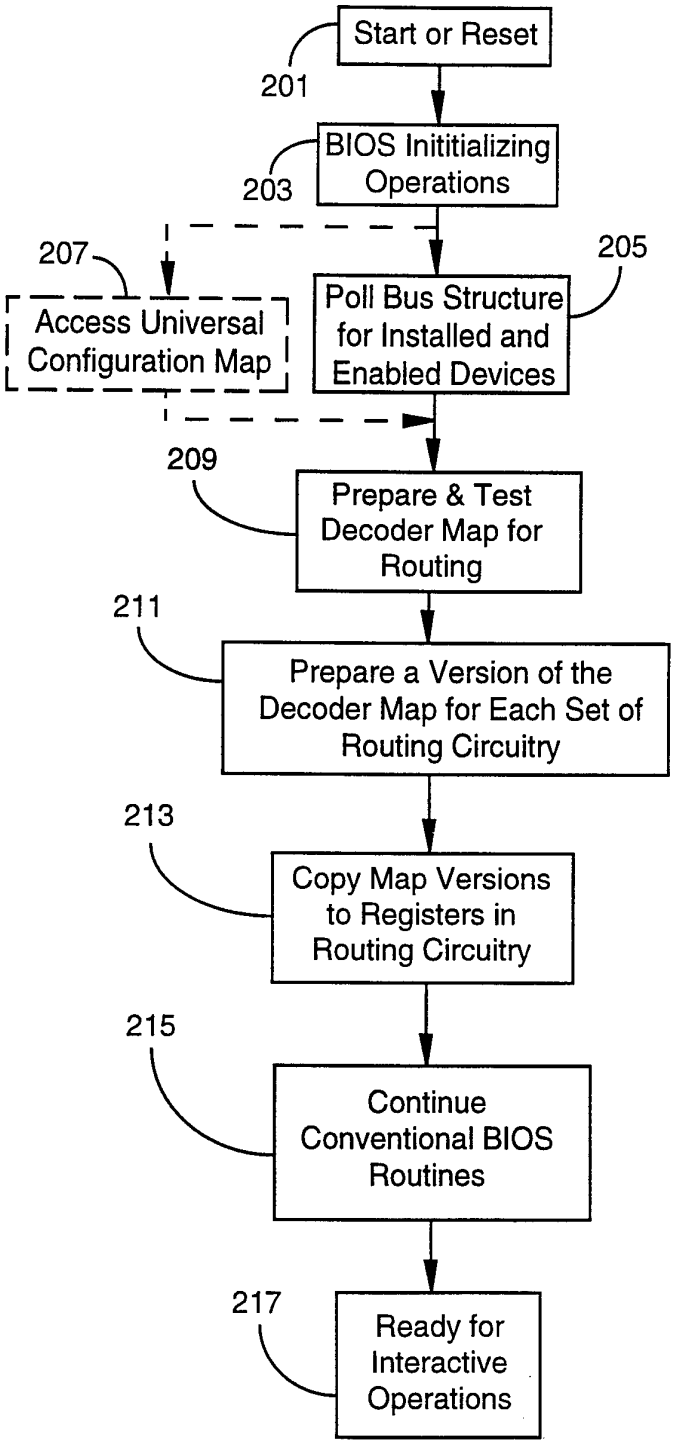

Start or Reset
201
BIOS Inititializing Operations
203
207
205
Access Universal Configuration Map
Poll Bus Structure for Installed and Enabled Devices
209
Prepare & Test Decoder Map for Routing
211
Prepare a Version of the Decoder Map for Each Set of Routing Circuitry
213
Copy Map Versions to Registers in Routing Circuitry
215
Continue Conventional BIOS Routines
217
Ready for Interactive Operations

Fig. 6

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US95/01951

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :G06F 9/445, 13/00
US CL :395/275; 364/238.2, 241
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)

U.S. : 395/275; 364/238.2, 241

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 practicable, search terms used)

APS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US, A, 5,247,682 (KONDOU ET AL) 21 September 1993, summary, col.3-col.4.	1-23
Y	US, A, 4,396,984 (VIDEKI, II) 02 August 1983, summary, col.4	4, 7-8
Y	US, A, 5,014,193 (GARNER ET AL.) 07 May 1991, col.2-5	1-23
Y	US, A, 4,688,172 (WRIGHT) 18 August 1987, abstract, description	1-23
Y,P	US, A, 5,305,437 (FRITZE ET AL) 19 April 1994, abstract, summary	9

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

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

"E" earlier document 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

Date of the actual completion of the international search
23 MARCH 1995

Date of mailing of the international search report
01/06/95

Name and mailing address of the ISA/US
Commissioner of Patents and Trademarks
Box PCT
Washington, D.C. 20231
Facsimile No. (703) 305-3230

Authorized officer
PARSHOTAM LALL
Telephone No. (703) 305-9715

Form PCT/ISA/210 (second sheet)(July 1992)★