PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ^{6} :
G06F 13/00, 13/14

A1

(11) International Publication Number: WO 95/16965

(43) International Publication Date: 22 June 1995 (22.06.95)

(21) International Application Number: PCT/US94/13702

(22) International Filing Date: 1 December 1994 (01.12.94)

(30) Priority Data:
08/168,463 16 December 1993 (16.12.93) US

(71) Applicant: INTEL CORPORATION [US/US]; 2200 Mission College Boulevard, Santa Clara, CA 95052 (US).

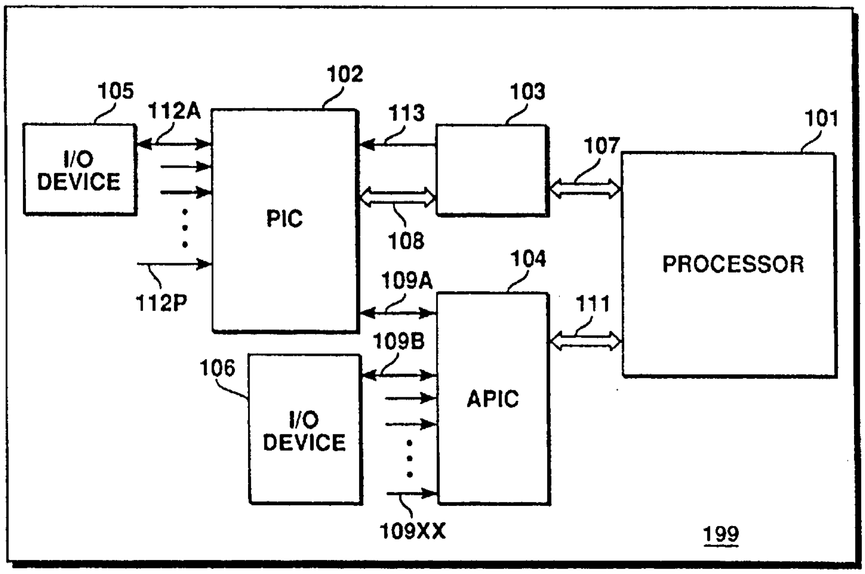
(72) Inventors: KARNIK, Milind; 7825 Southwest 186th Avenue, Aloha, OR 97007 (US). BATZ, Joseph; 8021 Southwest 159th Place, Beaverton, OR 97007 (US).

(74) Agents: BEREZNAK, Bradley, J. et al.; Blakely, Sokoloff, Taylor & Zafman, 7th floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025 (US).

(81) Designated States: AM, AT, AU, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, JP, KE, KG, KP, KR, KZ, LK, LR, LT, LU, LV, MD, MG, MN, MW, NL, NO, NZ, PL, PT, RO, RU, SD, SE, SI, SK, TJ, TT, UA, UZ, VN, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG), ARIPO patent (KE, MW, SD, SZ).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: MULTIPLE PROGRAMMABLE INTERRUPT CONTROLLERS IN A MULTI-PROCESSOR SYSTEM



(57) Abstract

An advanced programmable interrupt controller (APIC 104) provides interface between the processors (101), the PICs (102) and I/O devices (105). The APIC (104) sends an interrupt request data packet (fig. 2A and fig. 2B) with a first field (281) set to a procesor indentification number, a second field (282) set to a type of the device that sends interrupt request and a third field (283) set to an interrupt vector if the device sending the interrupt request to the APIC (104) is a device other than PIC (102) or to a predetermined identification number of the PIC (102) if the request is from the PIC (102). If the interrupt request is from a PIC (102), the processor (101) uses the third field (283) to identify which of the multiple PICs (102) caused the interrupt. If the interrupt is from a device, other than a PIC (102), the processor (101) uses the third field (283) to determine an interrupt vector corresponding to 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
AU	Australia
BB	Barbados
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BJ	Benin
BR	Brazil
BY	Belarus
CA	Canada
CF	Central African Republic
CG	Congo
CH	Switzerland
CI	Côte d'Ivoire
CM	Cameroon
CN	China
CS	Czechoslovakia
CZ	Czech Republic
DE	Germany
DK	Denmark
ES	Spain
FI	Finland
FR	France
GA	Gabon
GB	United Kingdom
GE	Georgia
GN	Guinea
GR	Greece
HU	Hungary
IE	Ireland
IT	Italy
JP	Japan
KE	Kenya
KG	Kyrgystan
KP	Democratic People's Republic of Korea
KR	Republic of Korea
KZ	Kazakhstan
LI	Liechtenstein
LK	Sri Lanka
LU	Luxembourg
LV	Latvia
MC	Monaco
MD	Republic of Moldova
MG	Madagascar
ML	Mali
MN	Mongolia
MR	Mauritania
MW	Malawi
NE	Niger
NL	Netherlands
NO	Norway
NZ	New Zealand
PL	Poland
PT	Portugal
RO	Romania
RU	Russian Federation
SD	Sudan
SE	Sweden
SI	Slovenia
SK	Slovakia
SN	Senegal
TD	Chad
TG	Togo
TJ	Tajikistan
TT	Trinidad and Tobago
UA	Ukraine
US	United States of America
UZ	Uzbekistan
VN	Viet Nam

MULTIPLE PROGRAMMABLE INTERRUPT CONTROLLERS IN A MULTI-PROCESSOR SYSTEM

FIELD OF INVENTION

The present invention pertains to the field of multi-processor computer systems and more specifically to input/output devices for use in such systems.

BACKGROUND OF THE INVENTION

Computer systems which utilize programmable interrupt controllers (PICs) in a single processor environment are well known in the prior art. Intel corporation's part number 8259A is an example of a PIC used in such single processor systems. The PICs serve as interface units between the input/output (I/O) devices and the processor in handling interrupt requests. The PIC accepts an interrupt request from the I/O devices and issues an interrupt request to the processor on an interrupt line. On receiving the interrupt request from the PIC, the processor generally initiates an interrupt acknowledge cycle. In the interrupt acknowledge cycle, the PIC sends an interrupt vector to the processor, which identifies an interrupt handler routine in the processor. The processor executes the interrupt handler routine to handle the interrupt pending on the I/O device.

With the advent of multi-processor computer systems, interrupt management systems that dynamically distribute the interrupt among the processors have been implemented. Intel's Advanced Programmable Interrupt Controller Architecture is an example of such a system. Typically, such an interrupt management system includes at least three functional components: 1. Advanced Programmable Interrupt Controllers (APIC) that provide interface lines on which I/O devices or PICs send interrupts into the system 2. Processors with modules to receive and service the interrupt requests from the I/O devices and 3. a bus that connects the processors and the APICs.

When an interrupt is pending, the I/O device or the PIC that is connected directly to the APIC sends an interrupt signal to the APIC. In response, the APIC broadcasts an interrupt request data packet on the bus. The interrupt request data packet contains a first field set to the identification number of the destination processor, a second field set to identify the type of device causing the interrupt and a third field set to the interrupt vector.

On receiving the interrupt request data packet, each of the processors examines the first field to determine if the interrupt request is directed to it. The processor then uses the interrupt vector in the third field to determine the address of the interrupt service routine. The processor executes the interrupt service routine to process the interrupt.

One of the basic requirements of a multi-processor system is to support PICs so as to be backward compatible with the prior application base. In this case, the PICs generally interface with the APICs using the interface lines of the APIC. The PIC is connected to the I/O devices and provides the interface between the APIC and these I/O devices. The APIC may have other I/O devices connected directly on the other interface lines.

When an I/O device connected to a PIC has an interrupt pending to be processed, it sends an interrupt request signal to the PIC. In response, the PIC sends another signal to the APIC on the interface lines. The APIC broadcasts an interrupt request data packet on the bus to which all the processors couple. In the interrupt request data packet, the first field is set to the identification number of processor to which the interrupt is to be directed to; and the second field is set to identify the type of device causing the interrupt. The APIC does not use the third field that is used to specify an interrupt vector. The processor, which processes the interrupt, gets the interrupt vector from the PIC in a subsequent interrupt acknowledge cycle. The processor determines an interrupt service routine based on the interrupt vector and processes the interrupt pending on the I/O device by executing the service routine.

However, these multi-processor based systems suffer from the drawback that only one PIC can be supported in a given multi-processor system. This is because there is no field in the interrupt request data packet to identify the PIC that initially caused the APIC to send the interrupt request data packet. What is needed is a mechanism to identify the PIC that has sent the interrupt request when multiple PICs are present in the multi-processor system so that the resulting interrupt acknowledge cycle can be directed to the identified PIC. This will enable the multi-processor system to support multiple PICs.

SUMMARY OF THE INVENTION

The present invention allows the operation of multiple programmable interrupt controllers (PIC) in a multi-processor system environment. The advanced programmable interrupt controllers (APIC) provide interface between the processors and the peripheral devices. The peripheral devices include a second I/O device and a PIC. The PIC further provides interface between a first I/O device and the APIC. The processors are connected to the APICs by means of a first bus.

The first I/O device sends an interrupt request signal to the PIC when the first I/O device has an interrupt pending to be processed. The PIC in response sends another interrupt request signal to the APIC. The APIC assembles an interrupt request data packet in response to the interrupt request signal from the PIC. The interrupt request data packet contains a third field set to an unique identification number of the PIC which had sent the interrupt request. The APIC sets the same third field to an interrupt vector corresponding to the second I/O device in response to an interrupt request from the second I/O device that directly interfaces with the APIC. The APIC sets a first field in the interrupt request data packet to an unique identification number of the destination processor. The APIC further sets a second field in the interrupt request data packet to a value corresponding to the type of the device that is sending the interrupt request signal.

The APIC broadcasts the interrupt request data packet on the first bus. The processor, to which the interrupt is directed to, receives the interrupt request data packet on the first bus. The processor examines the fields in the packet received to determine the type of device causing the interrupt.

If the type of device causing the interrupt is a PIC, the processor uses the PIC identification number in the third field to identify the specific PIC causing the interrupt. The processor initiates communication with the PIC to get an interrupt vector corresponding to the first I/O device that originated the interrupt request. If the type of device indicated is an I/O device, the processor gets the interrupt vector corresponding to the second I/O device from the third field in the interrupt request data packet. The processor identifies an interrupt processing routine based on the interrupt vector and processes the interrupt pending on the I/O device.

Thus, a multi-processor system that supports multiple PICs, as well as other I/O devices, is described.

BRIEF DESCRIPTION OF THE DRAWINGS

In the foregoing specification, the invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modification and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specifications and drawings are, accordingly, to be regarded illustrative rather than restrictive.

Figure 1 is a block diagram of a computer system in accordance with the present invention.

Figure 2A is the packet format used by the APIC to communicate with the processor when an interrupt request is received from a PIC.

Figure 2B is the packet format used by the APIC to communicate with the processor when an interrupt request is received from an I/O device other than a PIC.

Figure 3A is a flow chart depicting the steps for processing an interrupt from a device coupled to a programmable interrupt controller(PIC) in accordance with the present invention.

Figure 3B is a flow chart depicting the steps for processing an interrupt from a device coupled to the APIC in accordance with the present invention.

DETAILED DESCRIPTION

A multi-processor computer system that supports multiple programmable interrupt controllers (PIC) is described. Numerous specific details are set forth in this description in order to provide a thorough understanding of the invention. It will be obvious, however to one skilled in the art that these specific details need not be used to practice the present invention. In other instances, well-known system components and circuits have not been described in detail in order to avoid obscuring the present invention.

Figure 1 shows a multi-processor computer system 199 comprising a processor 101 coupled to an APIC 104 via a bus 111. Each of the processors has an unique processor identification number associated with it. A programmable interrupt controller PIC 102 and an I/O device 106 such as a disk drive use the interface lines 109-A and 109-B respectively to send an interrupt request to APIC 104. In one embodiment, APIC 104 supports 240 such interface lines. In a preferred embodiment, there can be more than one PIC interfacing with APIC 104. Similarly, there can be more than one APIC in a multi-processor computer system 199 in accordance with the present invention. Each of PICs in the multi-processor system has a unique PIC identification number associated with it.

The PIC 102 provides input lines 112-A to 112-P for receiving interrupt request signals from various I/O devices. For example, an I/O device 105 uses line 112-A to send an interrupt request to PIC 102. In response, PIC 102 sends an interrupt request signal to APIC 104 on interface line 109-A. On the other hand, I/O device 106 sends an interrupt request signal directly to APIC 104 when an interrupt is pending on I/O device 106. In response to an interrupt request from PIC 102 or I/O device 106, APIC 104 broadcasts an interrupt request data packet on bus 111. The processor 101 receives the interrupt request data packet on bus 111 and processes the interrupt if it is directed to it.

The multi-processor computer system further comprises a communication device 103 for providing interface between processor 101 and a PIC 102. The communication device 103 uses a bus 107 to communicate with processor 107. The communication device 103 uses an interrupt acknowledge line 113 to signal an interrupt acknowledge cycle to PIC. The PIC 102 uses bus 108 to send an interrupt vector, corresponding to the I/O device that initiated the interrupt process request to the communication device 103.

Figure 2a shows the format of the interrupt request data packet, the APIC 104 broadcasts on bus 111 in response to an interrupt request from PIC 102 on interface line 109-A. The packet comprises three fields. The first field specifies the destination processor to which the interrupt request is directed to. In a preferred embodiment, specific values are defined to specify a group of processors. In the case of a single processor being specified, the unique processor identification of the intended processor is used. The second field specifies the type of device that sent the interrupt request to the APIC 104. In a preferred embodiment a value of 7 specifies that a device of type PIC has sent the interrupt request to the APIC. However, it will be obvious to one skilled in the art the present invention can be practiced with a different set of values without departing from the spirit of the present invention. When the second field indicates that a device of type PIC sent the interrupt request, the third field specifies the PIC identification of the PIC 102 in the field. This field is subsequently used by communication device 103 to identify which of the multiple PICs actually sent the interrupt request to the APIC.

Similarly figure 2b shows the format of the interrupt request data packet the APIC 104 broadcasts in response to a interrupt request from I/O device 106 which interfaces directly to APIC 104. The first and second fields are used similar to in figure

2a. The first field specifies the destination processor. The second field specifies the type of the device that sent the interrupt request on the interface lines of APIC 104. In the case when the type of device is not a PIC, the third field is set to an interrupt vector corresponding to I/O device 106. The interrupt vector identifies a interrupt service routine in processor 101. In a preferred embodiment, the interrupt vector table is stored in APIC 104 and can be set by the user.

Figure 3A shows a flow-chart illustrating the processing of an interrupt on the I/O device 105 coupled to PIC 102 in accordance with the present invention. In step 202, I/O device 105 determines if there is a interrupt pending that needs to be processed. When an interrupt is pending, I/O device 105 sends an interrupt request signal to PIC 102 using line 112-A in step 203. In response, PIC 102 receives the interrupt request and sends an interrupt request signal to APIC 104 on line 109-A in step 204.

APIC 104 receives the interrupt request signal and assembles a interrupt request data packet in step 205 in accordance with format in Figure 3A. The first field is set to the unique identification number of the processor, if the interrupt is to be directed to a single processor. On the other hand, if a group identification number may also be specified to specify a group. The second field is set to a value 07 to indicate that a device of type PIC has sent the interrupt request to APIC 104. The third field is set to the PIC identification number of PIC 102.

APIC 104 broadcasts the packet thus assembled on bus 111 in step 205. All the processors in the multi-processor system, including the processor 101, receive the packet in step 206. Each of the processors examines the first field to see if the value in the first field matches the processor's own unique processor identification number in step 207. If there is a match, processor 101 determines that the interrupt is directed to it. If the destination address specifies a group of processors, the processors in the group may engage in an arbitration process to determine which processor should handle the interrupt. If processor 101 is determined not to be the one to handle the interrupt request, processor 101 discards the packets in step 208.

If processor 101 is determined to be the one to process the interrupt request, processor 101 forwards the third field in the interrupt request data packet to communication device 103 requesting to initiate an interrupt acknowledge cycle in step 209. This third field contains the PIC identification number of PIC 102 that sent the interrupt request signal to APIC 104. The communication device 103 uses the PIC identification number to identify PIC 102 among the multiple PICs present in the multi-processor system. The communication device 102 initiates an interrupt acknowledge cycle with the PIC thus determined in step 210. In step 211, the PIC sends and communication device 103 receives an interrupt vector corresponding to the interrupt pending on I/O device 105 coupled to PIC 102 that initiated the interrupt processing. The communication device 102 passes the interrupt vector to processor 101 in step 212. Processor 101 receives the interrupt vector and identifies the corresponding interrupt service routine in step 213. In a preferred embodiment, the interrupt vector is a pointer to the address of an interrupt service routine in the processor. Processor 101 then executes the interrupt service routine to process the interrupt pending on I/O device 105 that initiated the interrupt request.

Similarly figure 3B shows a flow chart describing the processing of an interrupt from I/O device 106 that interfaces directly to APIC 104. In step 252, I/O device 106 determines if there is interrupt pending that needs to be serviced by a processor in the multi-processor system. When it determines so, I/O device 106 sends a signal to APIC 104 in step 253.

On receiving the signal, APIC 104 assembles an interrupt request data packet in accordance with the format in Figure 2b. The first field is set to the processor identification number of the processor to which the interrupt is to be directed to. The field may also specify a value to specify a group of processors. The second field is set to a value to indicate the type of I/O device 106. The third field is set to an interrupt vector value corresponding to I/O device 106. As explained, the interrupt vector table is stored in APIC 104 in a preferred embodiment.

In step 254, APIC 104 broadcasts the interrupt request data packet assembled on bus 111. All the processors coupled to bus 111 receive the interrupt request data packet in step 255. Processor 101 then determines if the interrupt request is directed to it based on the value in first field of the packet in step 256. If the value corresponds to multiple processors, the processors engage in an arbitration scheme to determine which processor is to process the interrupt. If the processor determines that the interrupt request is not directed to it, it discards the interrupt request data packet.

The processor 101 determines that the type of I/O device 106 sending the interrupt request to the APIC is not a PIC based on the value in the second field in the interrupt request data packet. The processor gets the interrupt vector from the third field in the interrupt request data packet. The processor then uses the interrupt vector to determine the interrupt service routine. In the present embodiment, the interrupt vector field is a pointer to the memory address of interrupt service routine. In step 256, the processor processes the pending interrupt using the interrupt service routine thus determined.

Thus the present invention describes a multi-processor system in which multiple PICs are supported.

CLAIMS

What is claimed is:

1. A multi-processor system including at least one processor, said multi-processor system comprising:

a first bus coupled to said at least one processor;

a first interrupt controller coupled to said first bus, said first interrupt controller having a first predetermined identification number;

a second interrupt controller coupled to said first bus and said first interrupt controller;

a first I/O device coupled to said first interrupt controller, said I/O device sending a first interrupt request signal to said first interrupt controller when an interrupt is pending on said first I/O device, said first interrupt controller further sending a second interrupt request signal to said second interrupt controller in response to said first interrupt request signal; and

a second I/O device coupled to said second interrupt controller, said second I/O device sending a third interrupt request signal to said second interrupt controller when an interrupt request is pending on said second I/O device.

2. The multi-processor system of claim 1 wherein said second interrupt controller broadcasts a interrupt request data packet on said first bus in response to said second interrupt request signal, said interrupt request data packet comprising a third field set to said first predetermined identification number.

3. The multi-processor system of claim 2 wherein said second interrupt controller broadcasts said interrupt request data packet in response to said third

interrupt request signal, said third field being set to an interrupt vector corresponding to said second I/O device, said interrupt vector identifying a second interrupt service routine to process said interrupt pending on said second I/O device.

4. The multi-processor system of claim 3 wherein said interrupt request data packet further comprises:

a first field for specifying a destination processor; and

a second field for specifying a type of device coupled to said second interrupt controller that has caused said second interrupt controller broadcast said interrupt request data packet.

5. The multi-processor system of claim 4 wherein said at least one processor includes receiving means to receive said interrupt request data packet;

6. The multi-processor system of claim 5 wherein said processor includes examination means for examining said first, second and third fields in said interrupt request data packet received, said first field being used to determine if said packet is directed to said at least one processor, said examination means further examining said second field to determine if said type of device coupled to said second interrupt controller causing said second interrupt controller to broadcast said interrupt request data packet is a first interrupt controller.

7. The multi-processor system of claim 6 wherein said processor further comprises:

means to identify said first interrupt controller based on said third field when said examination means determines said type of device coupled to said second interrupt controller causing said second interrupt controller to broadcast said interrupt request data packet is a first interrupt controller; and

communication means for communicating with said first interrupt controller, said first interrupt controller sending and said communication means receiving a first interrupt vector, said first interrupt vector identifying a first interrupt service routine to process said pending interrupt.

8. The multi-processor system of claim 7 wherein said communication means comprises:

an external circuitry coupled to said first interrupt controller ; and

a second bus coupled to said processor and said external circuit;

9. The multi-processor system of claim 8 wherein said processor processes interrupt pending on said first I/O device based on said first interrupt service vector.

10. The multi-processor system of claim 9 wherein said interrupt request data packet is received by all processors coupled to said bus.

11. The multi-processor system of claim 10 wherein said third field can specify more than one processors.

12. The multi-processor system of claim 11 wherein said processor determines said second interrupt vector based on said third field if said examination means determines that said type of device is not a first interrupt controller, said processor identifying said second interrupt service routine based on said second interrupt vector and processing said interrupt pending on said second I/O device.

13. The multi-processor system of claim 12 wherein said at least one processor processes said interrupt pending on said first I/O device executing said interrupt service routine.

14. A method of processing interrupts in a multi-processor system including at least a first processor, comprising the steps of:

sending a first interrupt request signal from a first I/O device to a first interrupt controller when an interrupt is pending on said first I/O device;

sending a second interrupt request signal from said first interrupt controller to a second interrupt controller in response to said first interrupt request signal, said first interrupt controller having a predetermined identification number; and

assembling an interrupt request data packet in response to said second interrupt request signal in said second interrupt controller, said interrupt request data packet comprising a third field set to said predetermined identification number.

15. The method of processing an interrupt of claim 14 further comprising the steps of:

sending a third interrupt request signal from a second I/O device to said second interrupt controller when an interrupt is pending on said second I/O device; and

assembling said interrupt request data packet in response to said third interrupt request signal in said second interrupt controller, said third field being set to a second interrupt vector corresponding to said second I/O device.

16. The method of processing an interrupt of claim 15 further comprising the steps of:

setting a first field in said interrupt request data packet in said second interrupt controller to a processor identification number of said at least one processor;

setting a second field in said interrupt request data packet to a type of device sending the interrupt to said second interrupt controller; and

sending said packet from said second interrupt controller to said at least one processor.

17. The method of processing interrupt of claim 16 further comprising the steps of:

receiving said interrupt request data packet by said at least one processor;

determining if said interrupt request data packet is directed to said at least one processor based on said second field;

ignoring said interrupt request data packet if said interrupt request data packet is not directed to said at least one processor;

determining if type of device coupled to said second interrupt controller causing said interrupt request data packet to be sent is a first interrupt controller;

determining if said first interrupt controller sent said third interrupt request signal based on said third field if the interrupt request is determined to be of first interrupt controller type; and

obtaining a first interrupt vector from said first interrupt controller to said at least one processor.

18. The step of obtaining a vector of claim 17 further comprising the steps of:

initiating an interrupt acknowledge cycle between said at least one processor and said first interrupt controller; and

said first interrupt controller sending and said at least one processor receiving a first interrupt vector.

19. The method of claim 18 further comprises the step of processing said interrupt pending by said processor based on said first interrupt vector.

20. The method of processing interrupts of claim 17 further comprising the steps of:

determining a second interrupt vector based on said third field of said interrupt request data packet if type of device sending to interrupt request to said second interrupt controller is determined not to be a first interrupt controller;

processing interrupt pending on said second I/O device based on said second interrupt vector.

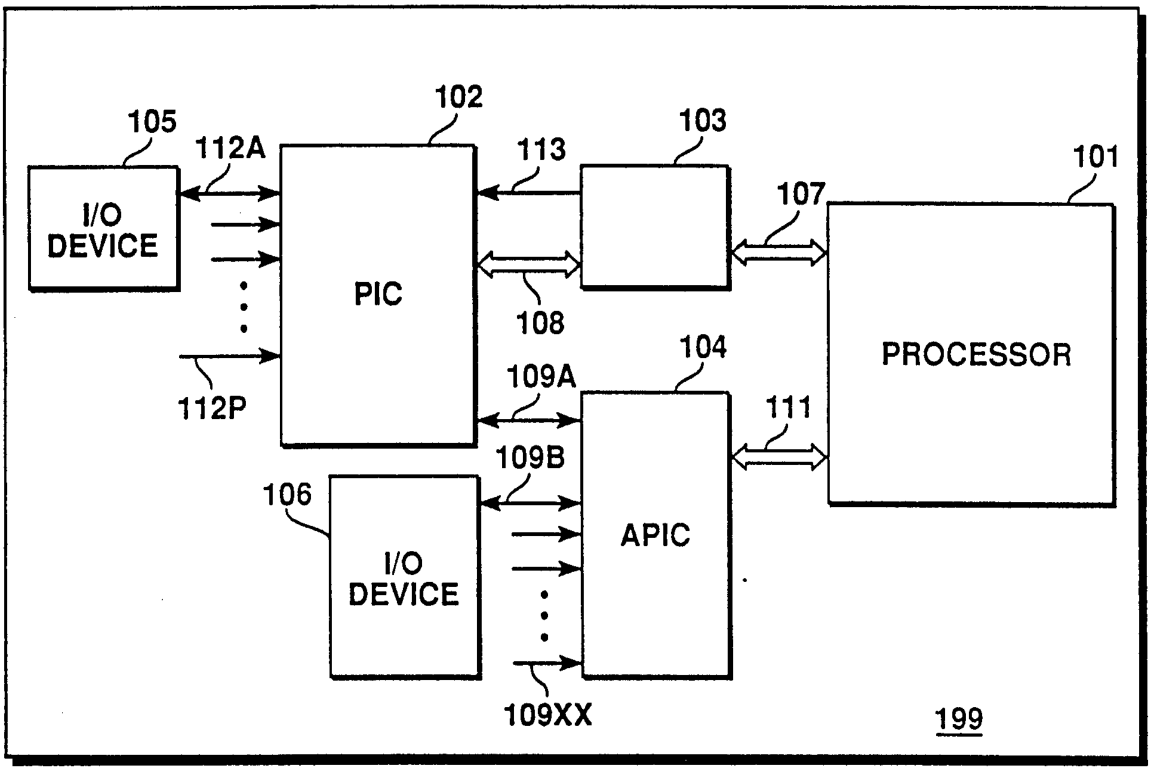
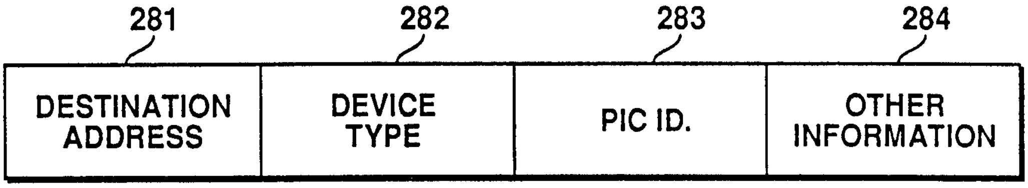
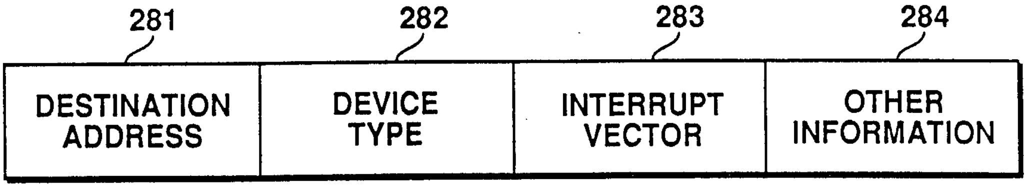


FIGURE 1



281	282	283	284
DESTINATION ADDRESS	DEVICE TYPE	PIC ID.	OTHER INFORMATION

FIGURE 2A



281	282	283	284
DESTINATION ADDRESS	DEVICE TYPE	INTERRUPT VECTOR	OTHER INFORMATION

FIGURE 2B

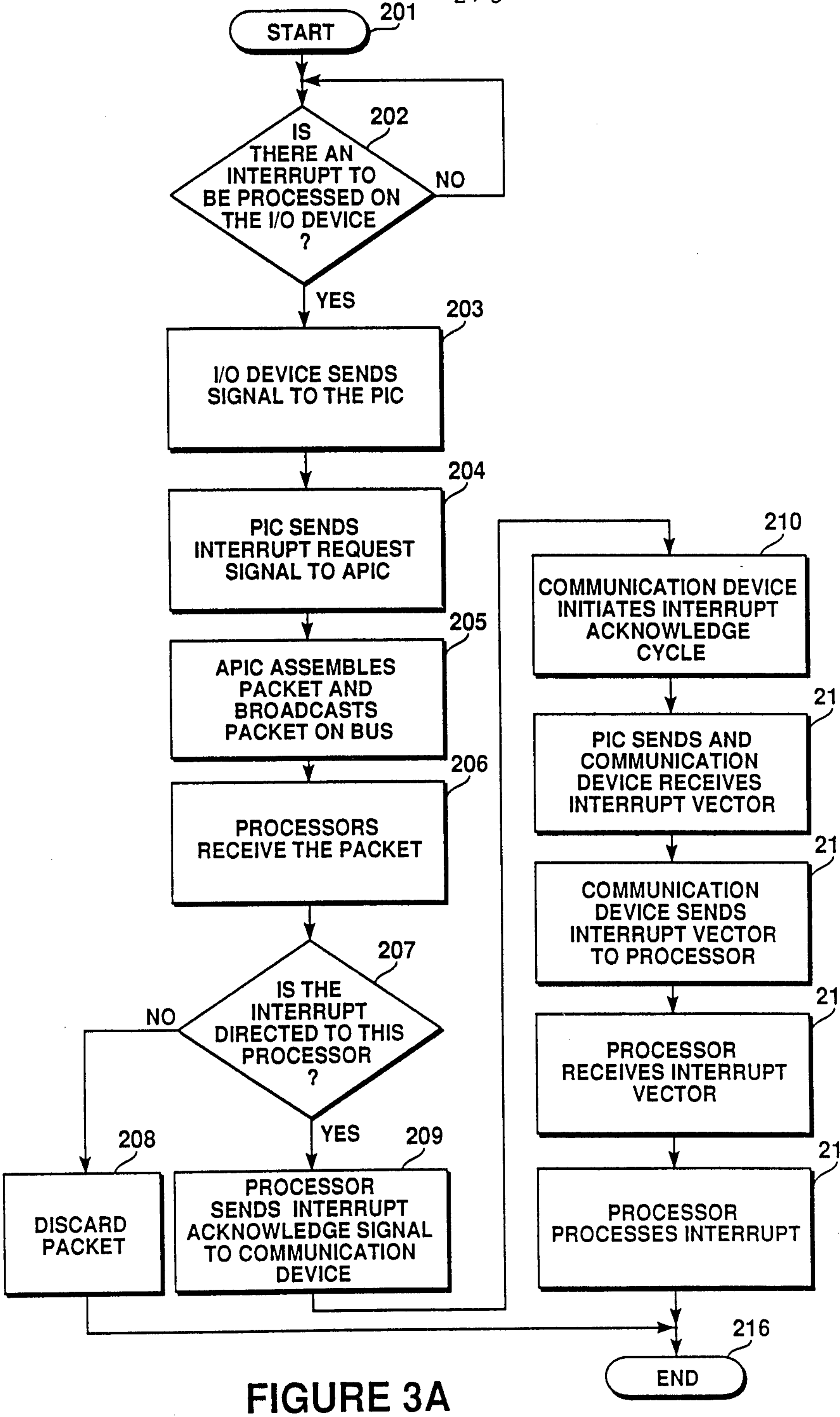

START
201
IS THERE AN INTERRUPT TO BE PROCESSED ON THE I/O DEVICE ?
202
NO
YES
203
I/O DEVICE SENDS SIGNAL TO THE PIC
204
PIC SENDS INTERRUPT REQUEST SIGNAL TO APIC
205
APIC ASSEMBLES PACKET AND BROADCASTS PACKET ON BUS
206
PROCESSORS RECEIVE THE PACKET
207
IS THE INTERRUPT DIRECTED TO THIS PROCESSOR ?
NO
YES
208
DISCARD PACKET
209
PROCESSOR SENDS INTERRUPT ACKNOWLEDGE SIGNAL TO COMMUNICATION DEVICE
210
COMMUNICATION DEVICE INITIATES INTERRUPT ACKNOWLEDGE CYCLE
211
PIC SENDS AND COMMUNICATION DEVICE RECEIVES INTERRUPT VECTOR
212
COMMUNICATION DEVICE SENDS INTERRUPT VECTOR TO PROCESSOR
213
PROCESSOR RECEIVES INTERRUPT VECTOR
214
PROCESSOR PROCESSES INTERRUPT
216
END

FIGURE 3A

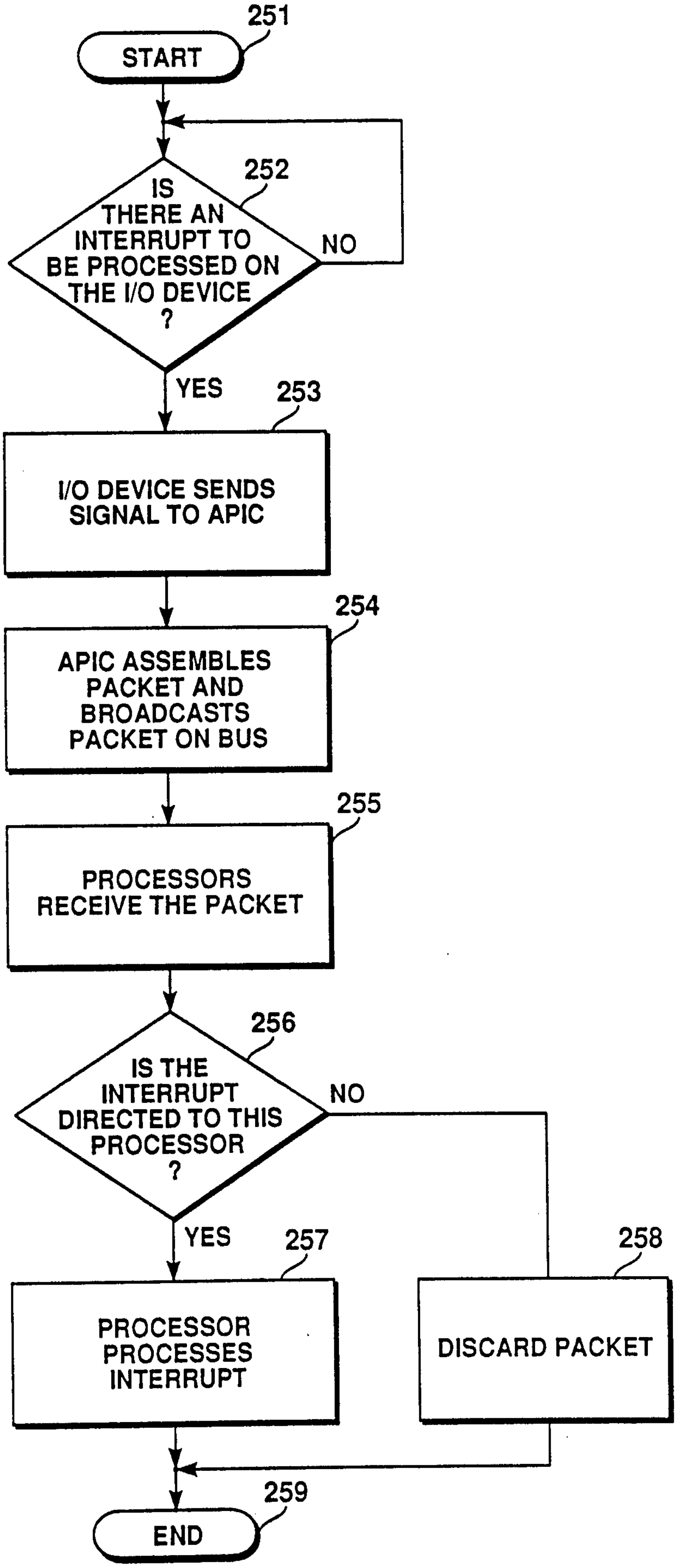

251
START
252
IS THERE AN INTERRUPT TO BE PROCESSED ON THE I/O DEVICE ?
NO
YES
253
I/O DEVICE SENDS SIGNAL TO APIC
254
APIC ASSEMBLES PACKET AND BROADCASTS PACKET ON BUS
255
PROCESSORS RECEIVE THE PACKET
256
IS THE INTERRUPT DIRECTED TO THIS PROCESSOR ?
NO
YES
257
PROCESSOR PROCESSES INTERRUPT
258
DISCARD PACKET
259
END

FIGURE 3B

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US94/13702

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :G06F 13/00, 13/14
US CL : 395/275, 725
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, 725

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)

Please See Extra Sheet.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	INTERFACING TO THE IBM PERSONAL COMPUTER, 1990, L.C. Eggebrecht, pp.150-153,	1-20
Y,P	US, A, 5,283,904 (CARSON ET AL.) 01 February 1994, col. 2-3	1-20
A	US, A, 4,271,468 (CHRISTENSEN ET AL.) 02 June 1981, Abstract.	1-20
A	US, A, 4,805,096 (CROHN) 14 February 1989, Abstract.	1-20

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

*	Special categories of cited documents:	"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
"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	"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
"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)	"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
"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	"&"	document member of the same patent family

Date of the actual completion of the international search 09 JANUARY 1995	Date of mailing of the international search report 09 MAY 1995
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 Thomas C. Lee Joni Hill Telephone No. (703) 305-9600

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US94/13702

B. FIELDS SEARCHED
Electronic data bases consulted (Name of data base and where practicable terms used):

```
APS
L1        258 S 364/230.2/CCLS
L2        974 S INTERRUPT CONTROLLER#
L3         30 S L1 AND L2
L4         91 S BROADCAST AND ID AND L2
L5        930 S INTERRUPT VECTOR#
L6         40 S L5 AND L4

=>
```